Volume: 5, Issue: 2
Page: 42-53
2021

International Journal of
Science and Business
Journal homepage: ijsab.com/ijsb

IJSAB International

Recourses Utilization in a Distributed System: A Review

Zhala Jameel Hamad & Subhi R. M. Zeebaree

Abstract:
Recourses in a distributed system can be accessed or remotely accessed across multiple nodes in the system. Nodes share resources including hardware, software, and data in distributed systems. A distributed system can always have a common purpose, such as addressing a significant computational issue. The set of independent processors is then interpreted by the user as a unit. And the distributed system's function is to organize the use of shared resources or to provide users with communication services. Therefore ,this paper focuses on resource utilization in different techniques and different domains that motivate and support a distributed system. Many previous works related to this subject have been addressed, explained in detail. The results showed that resources play a vital role in facilitating organizational processes, reduce work runtimes, and achieve a performance improvement; it was also beneficial for helping users' process massive data. The findings also explored that resource utilization aspects and factor's impact are varied from the distributed system.

IJSB
Literature Review
Accepted 20 January 2021
Published 25 January 2021
DOI: 10.5281/zenodo.4461813

Keywords: Resources utilization, distributed system, shared resources.

About Author (s)

Zhala Jameel Hamad (corresponding author), ISE Department, Erbil Polytechnic University, Erbil - Kurdistan Region - Iraq, zhalla.mei20@epu.edu.iq.
Subhi R. M. Zeebaree, Information Technology Department, Duhok Polytechni University, Duhok - Kurdistan Region – Iraq, subhi.rafeeq@dpu.edu.krd.

1. Introduction

A distributed system may consist of nodes connected via a communication network, possibly computing and heterogeneous nodes. All the resources of any node should be accessible transparently and conveniently from other nodes (Dino et al., 2020; Jader et al., 2019; Rashid et al., 2018). Heterogeneity is also another major evolution in distributed systems can take multiple forms (Haji et al., 2020; Zebari et al., 2018; Zeebaree et al., 2019; Zeebaree et al., 2020; Zhou, 1992). The fault tolerance in a distributed model is much higher than other network models, so the performance ratio is much better (Haji et al., 21, May; Harki et al., 2020; Shukur et al., 2020). The main goals in a distributed system are accessibility, transparency, trustworthiness, scalability, and performance (Alzakholi et al., 2020; Warneke & Kao, 2011; Zebari et al., 2019).Many distributed systems, with both private and public clouds as well as data centers, are shared by several tenants today. These include popular storage, data analytics, queuing, databases, or services such as Azure Storage, Amazon SQS, HDFS, or Hive for coordination in terms of elasticity and cost, multi-tenancy has clear benefits (Mace et al., 2015; Zeebaree et al., 2020; Zeebaree, et al., 2020). Large distributed computing systems such as clouds, grids, and voluntary computing systems typically comprise thousands to millions of computer units working across complex hierarchical, large-scale networks (Abdullah et al., 2020; Cheng & Agrawal, 2007; Dino et al., 2020.). In general, resources are very heterogeneous, inconsistent and shared by users and applications, making it incredibly hard to explain the behavior of certain systems. The study of such structures must generally address multiple observed entities, it uses monitoring tools to verify the successful use of such platforms. Some of these systems collect information from resources such as the use of the network or processor and provide general statistics created from traces of the use of basic resources (Sallow et al., 2020.; Schnorr et al., 2012; Zeebaree et al., 2020). Types of resources, computing resources (I/O device, memory, processor), storage resources, networking resources, power resources (Dino et al., 2020; Rashid et al., 2019; Zeebaree et al., 2019). There are numerous techniques to improve the energy efficiency of networking and computing resources, which are major elements of large-scale distributed systems. The solutions operate at various levels for computing resources, from nodes of individuals to entire infrastructures where recent advanced features such as virtualization are used (Barham et al., 2003; Zebari, 2011; Zeebaree et al., 2019). In parallel, shutdown methods for wired networks have been extensively studied and evaluated to reduce the amount of resources that can potentially stay idle and consume energy unnecessarily (Agarwal et al., 2010; Ibrahim et al., 2019; Zebari, 2013). Techniques are already available to modify the efficacy of network and computing resources to meet the needs of app providers and applications. In large-scale distributed systems, these techniques are also merged and executed in a structured manner (Jghef & Zeebaree, 2020; Orgerie et al., 2014; Zeebaree & Zebari, 2014). Distributed real-time embedded systems are commonly used in our lives. These systems can be seen in aircraft, industrial robots, vehicles, etc. functional (Abdullah et al., 2020; Anthony & Navghare, 2016). Therefore, this review paper focuses on resources and components that are utilized or organized in a distributed system in order to clarify the benefits of applying such a system in various fields.

2. Literature Survey

Anthony and Navghare (2016) given an understanding of the different sources of renewable energy utilized for distributed generation of power like (Solar Energy, Wind Energy, Bioenergy, Geothermal Energy, etc.) The distributed generation could be a clean process and decreases the unhealthy effects on the environment. As fossil fuels are restricted, the distributed era empowers the coherence of electrical power for the ever-increasing request and helps the country increment its differing qualities of energy sources and the economy remains smooth (Anthony & Navghare, 2016). Mayhorn et al. (2016) presented the joining energy storage and distributed energy resources (DERs), for instance, renewable resources, in control systems it can permit for more economical power services. The key challenge for confined control systems with high infiltration of DERs is that normal changes in load and renewable generation can be more critical compared to interconnected systems, Focusing on the kind of load and the extent of renewable energy infiltration. The extra wind control varieties typically impact control system operations on different time scales. A robust solution and a comprehensive formulation method were displayed for planning DERs in an isolated distribution system with high infiltration of renewable energy resources. The goals were to decrease the operating costs of diesel generators and increase the employment of wind obstetrics (Mayhorn et al., 2016). Radivilova et al. (2016) proposed a load balancing solution focused on measuring the load of distributed system nodes. When determining the load of the nodes, the average CPU and transmitting power, memory, are determined on the basis of the measured load by the accounting system or the monitor operating system. This allows the system variance, the duration of work and the efficiency of the usage of system resources to be measured. Using the proposed strategy requires requests to be spread over servers in order to decrease the deviation between the load servers and the average value, which permits to induce higher system execution measurements and quicker handling streams (Radivilova et al., 2016). Thamsen et al. (2016) proposed distributed structures of dataflow such as Spark or Flink. Allow big datasets to be analyzed by users. Users build programs Choose a collection of systems and resources by offering a set of well-defined operations with sequential user-defined functions. Distribute that employment automatically through these resources. Furthermore, inherently, the allocation of resources for particular performance requirements is difficult and, as a result, consumers appear to over-provision, which results in low utilization of clusters. Bell, a distributed submission tool data flow jobs that pick resources automatically in compliance with Targets for runtime. Bell models the scale-out actions for this, depending on past work executions for various jobs. Accordingly, in a staging cluster, isolated training runs are not needed for Bell. It also does not require processing instrumentation either, systems, or changes to the resource administration framework. As of now, Bell makes numerous users scale-out options for their runtime goals. Users should unambiguously state their execution targets and have Bell perform the resource allocation rather than effectively estimating the number of properties (Thamsen et al., 2016).

Kirichenko & Radivilova (2017) proposed a solution to the real scientific issue of load balancing and the efficient use of distributed system resources. The method proposed is based on, memory, bandwidth, and load CPU estimation, across flows of different support groups for each server and the distributed system as a whole. The multi-fractal property of the input is taken into account and flow of data. Factors for weighting have been introduced to determine the function of server characteristics relative to each other. This approach makes it possible, therefore, the imbalance of all system servers and the use of the system should be calculated. For different multifractal parameters, the simulation of the method suggested for the input flows. The simulation showed that the characteristics of multifractal traffic had a significant effect on the system imbalance. The use of the approach proposed helps to the distribution of requests through servers in such a way that the load server's deviation from the middle value has been less, which helps the system getting greater metrics efficiency and quicker flows of processing (Kirichenko & Radivilova, 2017). Cheng et al. (2017) proposed Auto-scaling which could be a strategy that allocates resources based on the variable workload. This paper centers on auto scaling with heterogeneous agreements for containers. The aim is to reduce the cost of container changes and reduce the insufficient penalty for the resource, while ensuring high usage of resources. Minimal costs are highly difficult to achieve without recognizing the future workloads ahead of time. Therefore, they suggested an optimal algorithm for dynamic programming that can give the potential workload, scale optimally. This model solution is being used as the basis for estimating various algorithms which do not have future workload data. At that point, they offered two covetous algorithms that don't require information workload in development, and a heuristic method which first predicts another time step's workload using gradient enhancing relapse, at that point makes scaling choices utilizing the optimal energetic programming algorithm. The experimentations demonstrate that when the cost is much higher for starting new servers,, Penalty for inadequacy resource, short term forecast the plan would only lift the overall cost by just 9.6% and reduce usage by a simple 10%, compared to Maximum dynamic programming, which knows the future workload (Cheng et al., 2017). Huang et al. (2017) displayed the joint optimization issue of load balancing and allocation of power in heterogeneous distributed implanted frameworks. From the hardware perspective, all nodes in the system are determined to be heterogeneous, with maximum speed and power consumption being different for each node. They also decide that each task's priority on each node is different and that each core's speed differs from the perspective of the application. To solve the joint optimization problem that used the Lagrange approach, they propose an efficient algorithm. By using an appropriate data system, they develop a system to calculate the suitable speed of and core to balance the correlation of core speed and task arrival rate when the problem cannot be fixed by the Lagrange method. The issue is resolved by this method. it makes an initial contribution to ideal allocation of power and load balancing with efficiency limitation for different implanted computing nodes in distributed embedded systems and heterogeneous (Huang et al., 2017). Thamsen et al. (2017) presented a resource enhancement approach Usage and job efficiency when scheduling recurring data in collaborative cluster environments, analysis jobs. The scheduler continuously learns using a reinforcement learning algorithm to classify how well different job combinations use resources when co-located are performed. Which works on the cluster are better done simultaneously. The evaluation of an execution built on Hadoop YARN appears that this approach can increment resource utilization and decrease work runtimes. Whereas interference between employments can be avoided, co-locations of employments with complementary resource utilization are not however continuously fully accepted (Thamsen et al., 2017)

Agrawal et al. (2017) have presented a cost-effective and dynamic proposed process of allocation of resources to manage large-scale streams of data from cameras that are distributed. Data obtained from millions of public cameras It is important to retrieve, store, and evaluate. It needs a system that requires large amounts of resources to be allocated for analysis of Visual data on a large scale. Shared storage is provided by cloud computing, computing, and numerous facilities to manage such a tremendous volume of data from distributed cameras obtained. In order to minimize the overall analysis cost, they presented a Resource allocation method that offers the degree of demand for cost-effective resources; automatic scaling in Proportion to the fluctuation of demand, within the cloud. They use the Nash Bargaining Method approach as the resource manager within the CAM2 component project. The efficiency of the model proposed has been confirmed by various tests in terms of the number of demands handled. Based on workload, the resource manager decides the bid value, the average Waiting time, and request from the user. Clearly, experimental effects Proved that the proposed algorithm can enhance performance and provide cost-effective use of resources over time (Agrawal et al., 2017). Kaufmann et al. (2018) presented that Modern distributed stacks of analytics consist of application frameworks that enable the processing of large quantities of data and a resource manager that allows applications to access machine resources. For these systems, the original use case was long-lived batch jobs (e.g., a few hours), but new use cases have emerged since their introduction, where people are increasingly using them to gain knowledge interactively or even online. In these additional use cases, the efficient sharing of resources involves smaller operating time scales (mins or maybe even secs) than the current systems been designed for and are able. They implemented Mira, a framework to optimize the elastic implementation of short-running and collaborative data analytics applications with the start-up of reduced execution, quick management of resources, and efficient use of resources on shared clusters. They evaluated the overhead sharing of resources in a widely used Parallel processing stack (Spark+YARN) and opportunities in shared environments to accelerate applications. Experiments show that Mira is capable of reducing the sharing of resources More than 400 associated overheads and reduces the runtime of applications by up to 4.2. (Kaufmann et al., 2018). Shan et al. (2018) suggested a new model of OS, named the split kernel, for the management of disaggregated systems. In order to increase resource efficiency elasticity, heterogeneity, and failure handling in data centers. In loosely-coupled monitors, the split Kernel disseminates. Traditional OS features, each running on and handling a hardware component. A split kernel often performs resource allocation and a distributed collection failure handling of hardware components. They created LegoOS, a modern OS for the disaggregation of equipment resources planned to utilize the split kernel demonstrate. LegoOS shows up for clients as a set of distributed servers. Internally, multiple processors, memory, and storage hardware components can span a user application. By imitating hardware components with servers' product, LegoOS was actualized and tried on x86-64. The results of the evaluation show that the efficiency of LegoOS is comparable to monolithic Linux servers (The monolithic model of a server, in view of several recent hardware and software developments, where the unit is a deployment, service and failure server, its limits are reached), Although enhancing resource packing significantly and reducing the rate of failure in monolithic clusters (Shan et al., 2018). Matsumura et al. (2018) presented a Virtual Power Plant (VPP), a hybrid technical and business plan for the implementation of a distributed resource and the advantages and possibilities of VPP are discussed. And mitigating the side effects of fluctuating energy in order to increase grid performance. Besides storage equipment such as mobile batteries and stationery, a VPP scheme was also proposed that includes cooperation of response to demand and distributed generation. Standard protocols like open ADR are required to establish independent owners of resources to easily allow 'plug and play' from an exchange level and a physical contact level in a contract with a VPP. Relevant examples and pilot tests performed in Japan are also presented with a view to validating the viability of the approach (Matsumura et al., 2018).

Chen et al. (2019) proposed and introduced a system to achieve data persistence and shuffling memory disaggregation, a framework that blends the Spark Big Data architecture with a modern in-memory distributed file system. The problem of optimizing performance at a reasonable cost is addressed by co-designing the proposed in-memory distributed file system with a large amount of persistent memory based on DIMM (PMEM) and RDMA technology. The disaggregation architecture enables separate scaling of each part of the system, which is especially useful for cloud deployments. The results of an empirical examination display that for shuffle-intensive apps with the same space for memory, the system can achieve up to 3,5-fold efficiency increase compared to Spark's default configuration. Also, by leveraging PMEM, the system can successfully increment the computing cluster's memory space at a fair cost, with an acceptable overhead of execution time concerning using only local DRAM (Chen et al., 2019). Intorruk & Numnonda (2019) calculated Apache Kafka, one of the earliest Big Data tools, and Apache Pulsar, which is identical to Kafka, and one of the newest Big Data tools, will compare the design, performance, and use of resources to manage big data in real-time. The results show which Pulsar performs in resource latency, throughput, and average usage, especially when the length of messages is small (such as 1 KB and 1MB) after both systems have been implemented in the same environment. The choice of Kafka or Pulsar relies on the aim of use.

If customers want a high-performance device and comfort for testing, they suggested that Kafka be used by users. Since Kafka was developed and used for a long time, Flume and Spark have also been incorporated with Kafka. Studying and improving Kafka would be simple because Kafka has several forums and users to help address the problems with other users (Intorruk & Numnonda, 2019). Haitao et al. (2019) applied the multi-objective Ant lion algorithm to further raise the rate of use of renewable energy, this study considered the allocation of reactive power compensation devices and power output systems within the planning of distributed power optimization setups. Establishing a multi-objective model of optimization in distributed generation with minimum construction costs, system for power storage and devices for reactive power compensation, minimal system transmission losses and optimum system voltage stability index. To solve the model of optimization, additionally, as an example, the IEEE-33 node distribution network system appears to be able to optimize the impact of distributed generation variability on the distribution network through the allocation approach proposed in this paper. And further, under the premise of maintaining system stability, increased the consumption rate of renewable energy (Haitao et al., 2019). Pandey et al. (2019) proposed a Brokering System to run large workflow applications for data processing within a budget using commercial cloud tools. The machine, instantiated as the host submission, is a GENI node. HTCondor is mounted on the submission host with command CyVerse, Pegasus. Besides, under Pegasus guidelines, an image analysis workflow is developed. Using Pegasus, the image analysis workflow is designed to operate on a specific collection of machines with identified IP addresses. The workflow is basically divided into subtasks by Pegasus While maintaining contact relations between the tasks, If the user initializes the workflow, Pegasus transmits HTCondor's jobs to plan computing resources. Based on the optimized configuration of a workflow with HTCondor, functions are computed locally or on specified computing machines. HTCondor then schedules and records the assignments to the remote computer node(s) in the same manner as local computer node jobs are currently scheduled. The workflow and output data are captured from CyVerse input data and generated in real-time and sent back to CyVerse storage. The outcome lets users of bioinformatics (i.e. investigators and educators) process large data in a user-friendly manner and cost-effective, additional compute nodes may be applied to the computing set in real-time (Pandey et al., 2019). Tuor et al. (2019) suggested a novel mechanism for forecasting and collecting the usage of resources in scalable way in a distributed system. The mechanism may be a close integration for algorithms of, temporal forecasting, adaptive transmission, and dynamic clustering. This lets each local node decide when to forward its current measurement to the central node in order to maintain the transmission rate below the limit value defined. The data center summarizes a limited number of clusters with the obtained data on the basis of measurements gathered from local nodes Submitted method of dynamic clustering to record the centroids and evolution of clusters. Models are trained from each cluster's time-varying centroids to forecast a set of future resource usage of local nodes as a good way to minimize the quantity of computation and time series forecasting. The effectiveness of the suggested mechanism is verified Using numerous real-world datasets through extensive experiments (Tuor et al., 2019). Wang et al. (2019) the distributed constrained optimization problem was studied in the presence of "small" and "large" delays by designing an algorithm for continuous-time. The algorithm switched delay was proposed to resolve this problem, dependent on switching strategies. By building functions for Lyapunov and giving some "large" delays restrictions, to guarantee empirical convergence, including its proposed algorithm, a necessary delay-dependent condition was already derived. The IEEE 118-bus framework was requested to substantiate the theoretical findings (Wang et al., 2019). Xiang et al. (2019) designed unicorn, the first centralized geo-distributed, multi-domain data analytics resource orchestration system. In Unicorn, via resource state abstraction with each field, they encode the availability of resources, Abstraction version of network view expanded to correctly represent the accessibility of various resources with decreased exposure to information using a set of linear inequalities .A novel, efficient cross-domain query method and an optimized resource information protocol that protects privacy were also designed to discover and incorporate reliable, minimum resource availability information for a collection of data analytics jobs across different domains. Furthermore, Unicorn also offers a complete resource orchestrator that measures optimum resource allocation decisions for jobs in data analytics. A Unicorn prototype has been implemented and the present preliminary assessment shows its usefulness and feasibility. (Xiang et al., 2019). Skondric et al. (2020) suggested that typically, simulations of the appropriate framework led to the development of distributed Peer to Peer storage systems. The rationale for system expansion is based on the outcomes of simulations. For the success of such simulations, the valid identification of input parameters is important. They obtained data on node presence and file generation patterns From the Network Faculty of Information Technology. A limited group of users with a similar pattern of behavior and the generation of files were found. They developed a user model based on analyzed data. Parameters for this user group, such as the average size of user files located at their storage locations, and the file generation pattern. After that, by using a user clustering strategy and coincidence matrix, the quality of distributed system storage was compared, the point of the research is to determine the moment of justification for using one of these two methods in peer determination. The simulation outcomes experiments showed that systems using the coincidence matrix had a higher degree of availability. Assuming that there was sufficiently shared disk space ready (Skondric et al., 2020).

3. Discussion

Each type of resource from a distributed system should be able to provide users with services at any time according to their requests. This paper concentrates on different resources for this reason. Models of use, scheduling algorithms, novel mechanisms, energy-saving algorithms, approaches to optimization of resource use, simulation of load balancing, learning algorithms, are discussed and evaluated. Each technique mentioned above has its own algorithm, tool/technique, objectives, and important outcomes, as shown in Table 2. After explaining all twenty researches browsed in Table 2, it can be recommended those resource utilization algorithms major and a significant element in the development of a distributed system.

Table 2: Comparison between previous studies of the proposed approaches.

Author	Year	Methodology	Objective	Significant Results
Anthony and Navghare (2016)	2016	different sources of renewable energy utilized for distributed generation of power	decreases the unhealthy effects on the environment	helps the country increment its differing qualities of energy sources and the economy remains smooth
Mayhorn et al. (2016)	2016	joining energy storage and distributed energy resources (DERs)	at the same time decrease operating costs of diesel generators and increase the employment of wind obstetrics	The extra wind control varieties typically impact control system operations on different time scales
Radivilova et al. (2016)	2016	Load balancing strategy	permits to produce higher system execution measurement s and quicker handling streams	distribute requests over servers become lower the deviation of the load servers from the average value.
Thamsen et al. (2016)	2016	Bell, a functional work-monitoring	Users will explicitly state their execution targets and have Bell	Scale-out decisions for their runtime goals are made by

		device Implementation	execute the allocation of resources.	various users. Instead of evaluating the number of assets effectively.
Kirichenko & Radivilova (2017)	2017	simulation of load balancing, input multifractal flows	To distribute requests across the servers.	Helps to achieve greater performance metrics and faster processing flows for the system.
Y.-L. Cheng et al. (2017)	2017	An auto-scaling strategy that apportions resources according to dynamic workload.	reduce the container cost adjustments and decrease the r esource inadequate penalty whe reas keeping up high resource utilization.	the plan would only lift the overall cost by only 9.6 percent and Reduce usage by just 10 percent
Huang et al. (2017)	2017	Lagrange theory and data fitting method	First, to obtain center speed, and the Lagrange approach illuminates the load balancing schedule after that.	power allocation for many embedded computing nodes with performance constraints.
Thamsen et al. (2017)	2017	reinforcement learnin g algorithm	classify how well different job c ombinations use resources whe n co-located are performed	increment resource utilization and Reduce runtime for job
Agrawal et al. (2017)	2017	Nash Bargaining Method	to minimize the overall analysis cost	can enhance performance and provide cost-effective use of resources over time
Kaufmann et al. (2018)	2018	Mira system	optimized elastic execution of short-running	reducing the sharing of resources More than 400 associated overheads and reduces the runtime of applications up for 4.2
Shan et al. (2018)	2018	operating system (LegoOS)	To enhance the use of resources in data centers, heterogeneity, elasticity, and failure handling	Improving the Packing of Resources and Reducing the rate of failure in monolithic clusters.
Matsumura et al. (2018)	2018	Virtual Power Plant (VPP) approach	To enhance the efficacy of the grid, mitigating the side effects of resource fluctuations	Creating independent owners of resources to easily make 'plug and play'.
Chen et al. (2019)	2019	persistent memory (PMEM) and RDMA technology	to achieve data persistence and shuffling memory disaggregation	the system can achieve up to 3.5-fold improvement in performance. and increasing the computer cluster's memory capacity at an affordable price, with a suitable execution time
Intorruk & Numnonda (2019)	2019	(Apache Kafka) and (Apache Pulsar)	to manage big data in real-time	Pulsar outperforms in latency, throughput, and average use of resources
Haitao et al. (2019)	2019	AntLion Optimizer(ALO) algorithm	Improving the rate of use of renewable energy, allocating devices for reactive power compensation and energy storage systems.	Enhance the impact on the distribution network of distributed generation variability and increase the usage rate of renewable energy.
Pandey et al. (2019)	2019	Brokering Framework	Using commercial cloud tools to run massive data processing within a budget.	Support users in a cost-effective and user-friendly way to process large data.
Tuor et al. (2019)	2019	novel mechanism and (dynamic clustering, and temporal forecasting, adaptive transmission)	to decrease the amount of computation and time-series forecasting	The effectiveness of the suggested mechanism is verified

		methods		
Wang et al. (2019)	2019	continuous-time algorithm and switched delay algorithm	to address the presence of "small" and "large" delays	guarantee the exponential convergence
Xiang et al. (2019)	2019	Unicorn framework	Multi-domain, geo-distributed Data Analytics.	Efficiency and feasibility of jobs across various domains and evaluation once implemented.
Skondric et al. (2020)	2020	(user model, User clustering) strategy coincidence matrix	To determine the moment of justification for using one of these two strategies in the determination of peers.	systems using the coincidence matrix has a higher degree of availability.

4. Conclusion

A distributed system is one way that can link the long-distance devices and heterogeneous components together in order to obtain and provide users with services. In general, the major challenge of using resources in a distributed system is to allow users to access a huge amount of data further to reducing time stimulation, and to improve efficiency with performance. We presented a number of methods and techniques with the different tools for resource utilization in the distributed system after all those papers have been checked.it can be concluded that resource utilization in each of the techniques has results and an objective that differs on another technique from various papers. But in the end, each technique proves the importance and necessity of resources used in a distributed system.

5. References

Abdullah, P. Y., Zeebaree, S. R., Jacksi, K., & Zeabri, R. R. (2020). AN HRM SYSTEM FOR SMALL AND MEDIUM ENTERPRISES (SME) S BASED ON CLOUD COMPUTING TECHNOLOGY. *International Journal of Research-GRANTHAALAYAH*, *8*(8), 56–64.

Agarwal, Y., Savage, S., & Gupta, R. (2010). Sleepserver: A software-only approach for reducing the energy consumption of pcs within enterprise environments. *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, 22–22.

Agrawal, B., Surbiryala, J., & Rong, C. (2017). Resource allocation in cloud-based distributed cameras. *2017 IEEE International Congress on Big Data (BigData Congress)*, 153–160.

Alzakholi, O., Haji, L., Shukur, H., Zebari, R., Abas, S., & Sadeeq, M. (2020). Comparison Among Cloud Technologies and Cloud Performance. *Journal of Applied Science and Technology Trends*, *1*(2), 40–47. https://doi.org/10.38094/jastt1219

Anthony, R. N., & Navghare, S. P. (2016). An insight to distributed generation of electrical energy from various renewable sources. *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS)*, 341–344.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., & Warfield, A. (2003). Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, *37*(5), 164–177.

Chen, S., Wang, W., Wu, X., Fan, Z., Huang, K., Zhuang, P., Li, Y., Rodero, I., Parashar, M., & Weng, D. (2019). Optimizing Performance and Computing Resource Management of in-memory Big Data Analytics with Disaggregated Persistent Memory. *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*.

Cheng, Y., & Agrawal, D. P. (2007). An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. *Ad Hoc Networks*, *5*(1), 35–48.

Cheng, Y.-L., Lin, C.-C., Liu, P., & Wu, J.-J. (2017). High resource utilization auto-scaling algorithms for heterogeneous container configurations. *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, 143–150.

Dino, H., Abdulrazzaq, M. B., Zeebaree, S. R., Sallow, A. B., Zebari, R. R., Shukur, H. M., & Haji, L. M. (2020). Facial Expression Recognition based on Hybrid Feature Extraction Techniques with Different Classifiers. *TEST Engineering & Management*, *83*, 22319–22329.

Dino, H. I., Zeebaree, S. R., Ahmad, O. M., Shukur, H. M., Zebari, R. R., & Haji, L. M. (2020). Impact of Load Sharing on Performance of Distributed Systems Computations. *International Journal of Multidisciplinary Research and Publications (IJMRAP)*, *3*(1), 30–37.

Dino, H. I., Zeebaree, S. R., Salih, A. A., Zebari, R. R., Ageed, Z. S., Shukur, H. M., Haji, L. M., & Hasan, S. S. (2020). *Impact of Process Execution and Physical Memory-Spaces on OS Performance*. *Technology Reports of Kansai University*, *62*(5), 2391–2401.

Haitao, L. I. U., Lun, X. U., ZHANG, C., Xiao, S. U. N., & Jing, C. (2019). Optimal allocation of distributed generation based on multi-objective ant lion algorithm. *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, 1455–1460.

Haji, L. M., Zeebaree, S. R., Ahmed, O. M., Sallow, A. B., Jacksi, K., & Zeabri, R. R. (2020). Dynamic Resource Allocation for Distributed Systems and Cloud Computing. *TEST Engineering & Management*, *83*(May/June 2020), 22417–22426.

Haji, L., Zebari, R. R., R. M. Zeebaree, S., Abduallah, W. M., M. Shukur, H., & Ahmed, O. (21, May). GPUs Impact on Parallel Shared Memory Systems Performance. *International Journal of Psychosocial Rehabilitation*, *24*(08), 8030–8038. https://doi.org/10.37200/IJPR/V2418/PR280814

Harki, N., Ahmed, A., & Haji, L. (2020). CPU Scheduling Techniques: A Review on Novel Approaches Strategy and Performance Assessment. *Journal of Applied Science and Technology Trends*, *1*(2), 48–55.

Huang, J., Li, R., An, J., Ntalasha, D., Yang, F., & Li, K. (2017). Energy-efficient resource utilization for heterogeneous embedded computing systems. *IEEE Transactions on Computers*, *66*(9), 1518–1531.

Ibrahim, B. R., Zeebaree, S. R., & Hussan, B. K. (2019). Performance Measurement for Distributed Systems using 2TA and 3TA based on OPNET Principles. *Science Journal of University of Zakho*, *7*(2), 65–69.

Intorruk, S., & Numnonda, T. (2019). A Comparative Study on Performance and Resource Utilization of Real-time Distributed Messaging Systems for Big Data. *2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 102–107.

Jader, O. H., Zeebaree, S. R., & Zebari, R. R. (2019). A State Of Art Survey For Web Server Performance Measurement And Load Balancing Mechanisms. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, *8*(12), 535–543.

Jghef, Y. S., & Zeebaree, S. R. (2020). State of Art Survey for Significant Relations between Cloud Computing and Distributed Computing. *International Journal of Science and Business*, *4*(12), 53–61.

Kaufmann, M., Kourtis, K., Schuepbach, A., & Zitterbart, M. (2018). Mira: Sharing Resources for Distributed Analytics at Small Timescales. *2018 IEEE International Conference on Big Data (Big Data)*, 231–241.

Kirichenko, L., & Radivilova, T. (2017). Analyzes of the distributed system load with multifractal input data flows. *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, 260–264.

Mace, J., Bodik, P., Fonseca, R., & Musuvathi, M. (2015). Retro: Targeted resource management in multi-tenant distributed systems. *12th ${$USENIX$}$ Symposium on Networked Systems Design and Implementation (${$NSDI$}$ 15)*, 589–603.

Matsumura, K., Marmiroli, M., & Tsukamoto, Y. (2018). *Smart grid for distributed resources and demand response integration*.

Mayhorn, E., Xie, L., & Butler-Purry, K. (2016). Multi-time scale coordination of distributed energy resources in isolated power systems. *IEEE Transactions on Smart Grid*, *8*(2), 998–1005.

Orgerie, A.-C., Assuncao, M. D. de, & Lefevre, L. (2014). A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys (CSUR)*, *46*(4), 1–31.

Pandey, A., Wang, S., & Calyam, P. (2019). Data-intensive Workflow Execution using Distributed Compute Resources. *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, 1–2.

R. M. Zeebaree, S., Haji, L. M., Rashid, I., Zebari, R. R., Ahmed, O. M., Jacksi, K., & Shukur, H. M. (2020). Multicomputer Multicore System Influence on Maximum Multi-Processes Execution Time. *TEST Engineering & Management*, *83*(May-June 2020), 14921–14931.

Radivilova, T., Kirichenko, L., & Ivanisenko, I. (2016). Calculation of distributed system imbalance in condition of multifractal load. *2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T)*, 156–158.

Rashid, Z. N., Zebari, S. R., Sharif, K. H., & Jacksi, K. (2018). Distributed Cloud Computing and Distributed Parallel Computing: A Review. *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 167–172.

Rashid, Z. N., Zeebaree, S. R., & Shengul, A. (2019). Design and Analysis of Proposed Remote Controlling Distributed Parallel Computing System Over the Cloud. *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 118–123.

Sallow, A. B., Sadeeq, M. A., Zebari, R. R., Abdulrazzaq, M. B., Mahmood, M. R., Shukur, H. M., & Haji, L. M. (2020). An Investigation for Mobile Malware Behavioral and Detection Techniques Based on Android Platform. *IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 22(4), 14-20.*

Schnorr, L. M., Legrand, A., & Vincent, J.-M. (2012). Detection and analysis of resource usage anomalies in large distributed systems through multi-scale visualization. *Concurrency and Computation: Practice and Experience*, *24*(15), 1792–1816.

Shan, Y., Huang, Y., Chen, Y., & Zhang, Y. (2018). Legoos: A disseminated, distributed ${$OS$}$ for hardware resource disaggregation. *13th ${$USENIX$}$ Symposium on Operating Systems Design and Implementation (${$OSDI$}$ 18)*, 69–87.

Shukur, H., Zeebaree, S., Zebari, R., Zeebaree, D., Ahmed, O., & Salih, A. (2020). Cloud Computing Virtualization of Resources Allocation for Distributed Systems. *Journal of Applied Science and Technology Trends*, *1*(3), 98–105.

Skondric, G., Hamulic, I., & Mudnic, E. (2020). Optimization of availability and resource utilisation in LAN based P2P storage distributed systems. *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 1–6.

Subhi R. M. Zebari, N. O. Y. (2011). Effects of Parallel Processing Implementation on Balanced Load-Division Depending on Distributed Memory Systems. *J. of University of Anbar for Pure Science*, *5*(3), Article 3.

Subhi Rafeeq Mohammed Zebari, A. S. Y. (2013). Improved Approach for Unbalanced Load-Division Operations Implementation on Hybrid Parallel Processing Systems. *Journal of University of Zakho*, *1*((A) No.2), Pp832-848.

Thamsen, L., Rabier, B., Schmidt, F., Renner, T., & Kao, O. (2017). Scheduling recurring distributed dataflow jobs based on resource utilization and interference. *2017 IEEE International Congress on Big Data (BigData Congress)*, 145–152.

Thamsen, L., Verbitskiy, I., Schmidt, F., Renner, T., & Kao, O. (2016). Selecting resources for distributed dataflow systems according to runtime targets. *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, 1–8.

Tuor, T., Wang, S., Leung, K. K., & Ko, B. J. (2019). Online Collection and Forecasting of Resource Utilization in Large-Scale Distributed Systems. *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 133–143.

Wang, X.-F., Hong, Y., Sun, X.-M., & Liu, K.-Z. (2019). Distributed optimization for resource allocation problems under large delays. *IEEE Transactions on Industrial Electronics*, *66*(12), 9448–9457.

Warneke, D., & Kao, O. (2011). Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud. *IEEE Transactions on Parallel and Distributed Systems*, *22*(6), 985–997.

Xiang, Q., Wang, X. T., Zhang, J. J., Newman, H., Yang, Y. R., & Liu, Y. J. (2019). Unicorn: Unified resource orchestration for multi-domain, geo-distributed data analytics. *Future Generation Computer Systems*, *93*, 188–197.

Zebari, R. R., Zeebaree, S. R., & Jacksi, K. (2018). Impact Analysis of HTTP and SYN Flood DDoS Attacks on Apache 2 and IIS 10.0 Web Servers. *2018 International Conference on Advanced Science and Engineering (ICOASE)*, 156–161.

Zebari, R., Zeebaree, S., Jacksi, K., & Shukur, H. (2019). E-Business Requirements for Flexibility and Implementation Enterprise System: A Review. *International Journal of Scientific & Technology Research*, *8*, 655–660.

Zeebaree, S. R., Jacksi, K., & Zebari, R. R. (2020). Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers. *Indonesian Journal of Electrical Engineering and Computer Science*, *19*(1), 510–517.

Zeebaree, S. R. M., M. Shukur, H., M. Haji, L., Zebari, R. R., Jacksi, K., & M.Abas, S. (2020). Characteristics and Analysis of Hadoop Distributed Systems. *Technology Reports of Kansai University*, *62*(4), 1555–1564.

Zeebaree, S. R., Sallow, A. B., Hussan, B. K., & Ali, S. M. (2019). Design and Simulation of High-Speed Parallel/Sequential Simplified DES Code Breaking Based on FPGA. *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 76–81.

Zeebaree, S. R., Shukur, H. M., & Hussan, B. K. (2019). Human resource management systems for enterprise organizations: A review. *Periodicals of Engineering and Natural Sciences*, *7*(2), 660–669.

Zeebaree, S. R., Zebari, R. R., & Jacksi, K. (2020). Performance analysis of IIS10.0 and Apache2 Cluster-based Web Servers under SYN DDoS Attack. *TEST Engineering & Management*, *83*(March-April 2020), 5854–5863.

Zeebaree, S. R., Zebari, R. R., Jacksi, K., & Hasan, D. A. (2019). Security Approaches For Integrated Enterprise Systems Performance: A Review. *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, *8*(12).

Zeebaree, S., & Zebari, I. M. (2014). Multilevel Client/Server Peer-to-Peer Video Broadcasting System. *International Journal of Scientific & Engineering Research*, *5*(8), 260–265.

Zhou, S. (1992). Lsf: Load sharing in large heterogeneous distributed systems. *I Workshop on Cluster Computing*, *136*.

Cite this article:

Zhala Jameel Hamad & Subhi R. M. Zeebaree (2021). Recourses Utilization in a Distributed System: A Review. ***International Journal of Science and Business,*** *5*(2), 42-53. doi: https://doi.org/10.5281/zenodo.4461813

Retrieved from http://ijsab.com/wp-content/uploads/669.pdf

Published by

IJSAB International