QALAAI ZANISTSCIENTIFIC JOURNAL
A Scientific Quarterly Refereed Journal Issued by Lebanese French University – Erbil, Kurdistan, Iraq
Vol. (6), No (4), Autumn 2021
ISSN 2518-6566 (Online) - ISSN 2518-6558 (Print)

Impact of Distributed-Memory Parallel Processing Approach on Performance Enhancing of Multicomputer-Multicore Systems: A Review

Dildar Masood Abdulqader
Department of Information Technology, Akre Technical College, Duhok Polytechnic University, Duhok, Kurdistan Region, Iraq
dildar.masood@dpu.edu.krd

Subhi R. M. Zeebaree
Culture Centre, Duhok Polytechnic University, Duhok, Kurdistan Region, Iraq.
subhi.rafeeq@dpu.edu.krd

ARTICLE INFO

Article History:
Received: 2/10/2021
Accepted: 16/11/2021
Published: Autumn 2021

Keywords:
Multicomputer, multicore, distributed system, parallel processing, performance enhancing.

Doi:
10.25212/lfu.qzj.6.4.45

ABSTRACT

Distributed memory is a term used in computer science to describe a multiprocessor computer system in which each processor has its own private memory. Computational jobs can only work with local data, so if you need remote data, you'll have to communicate with one or more remote processors. Parallel and distributed computing are frequently used together. Distributed parallel computing employs many computing devices to process tasks in parallel, whereas parallel computing on a single computer uses multiple processors to execute tasks in parallel. Distributed systems are designed separately from the core network. There are different kinds of distributed systems such as peer-to-peer (P2P) networks, groups, grids, distributed storage systems. The multicore processor can be classified into two types: homogeneous and heterogeneous. This paper reviews the impact of the distributed-memory parallel processing approach on performance-enhancing of multicomputer-multicore systems. Also, number of methods have been introduced which used in distributed-memory systems and discuss which method is the best and enhance multicore performance in distributed systems. The best methods were those which used an operating system named gun/Linux 4.8.0-36, intel Xeon 2.5, python programming language.

1. Introduction

A distributed system is a group of standalone computers that appear to their users as one coherent system. Distributed computing uses many remote computers geographically and solves large and complex tasks with high efficiency (Agarwal 2004)(Dino et al. n.d.). However, when deployed through distributor applications today in data centres, it is possible to participate in the design of distributed systems with their own network layer, but can offer significant benefits (Ports et al. 2015). Distributed systems display the best price and performance than concentrate systems(Van Steen and Tanenbaum 2017). Computer power can be added in a tiny boost to distributed systems. Distributed systems allow multiple users to access a shared computing resource, providing resource sharing (Bansal, Sharma, and Trivedi 2011)(Shukur et al. 2020).

Multicore processors have become popular and have a compact parallel computing power that cannot be completely used unless the program in progress is written accordingly. Writing an effective, scalable parallel program is very complex. In order to get the most out of multi-core processors, many parallels are needed in order to effective implementation of a program on a higher number of cores effectively. Simultaneous execution of several programs on several cores (Bridges et al. 2008) (McCool 2008). Multi-core processors have been around since the last decade, but they gained importance later on due to the limitations of technology that single-core processors face today(Wang et al. 2010) such as high throughput and long battery life with high energy efficiency(Ramanathan 2006). Multicore processors can be implemented in many ways depending on the application requirements. It can be implemented either as a heterogeneous group, as a homogenous group and a mixture of the two(Blake, Dreslinski, and Mudge 2009).

This paper introduces the performance-enhancing of multicomputer-multicore distributed memory systems based on parallel processing principles in section 2 and section 3. Section 4 gives a literature review about performance-enhancing of multicomputer-multicore distributed memory systems based on parallel processing principles and discussion of the surveyed methods. Section 5 is the conclusion of the review paper.

2. Distributed system

Distributed systems are designed separately from the core network, making the worst assumptions (Ports et al. 2015). Furthermore, the issue of operating systems display in distributed systems proposes new solutions to existing problems (Lu et al. 2016) (Haji et al. 2020). Distributed systems allow multiple users to access a shared computing resource, providing resource sharing. Examples of distributed computing include online rail reservation systems, air traffic control, internet banking, etc.(Bansal et al. 2011).

a. Peer-to-Peer (P2P) Distributed Systems

P2P is a class of application that benefits from the storage of resources, courses, content, and human existence that are accessible on the Internet frontier. Since recovering these decentralized resources means working in an unstable connections environment and unpredicted IP addresses, P2P nodes need to work outside of DNS and have worthy or complete independence for intermediate servers(Wang and Li 2003).

The existing Peer-to-Peer (P2P) networks can be categorized into three generations: First generation: The first P2P networks, such as napster and gnutella, aim to spread easily and quickly. Second generation: P2P networks typically use DHT technology to achieve better scalability and better query efficiency, and provide load balancing and inevitable search guarantees. Third generation: The recently proposed P2P network aims to provide high flexibility assuming that the node will collapse with some possibility of failure(Fiat and Saia 2002). The structure means that the topology of the P2P network is tightly controlled (such as Mesh(Zhao, Kubiatowicz, and Joseph 2001)(Rowstron and Druschel 2001), Ring(Stoica et al. 2001), d-dimension Torus(Ratnasamy et al. 2001)).

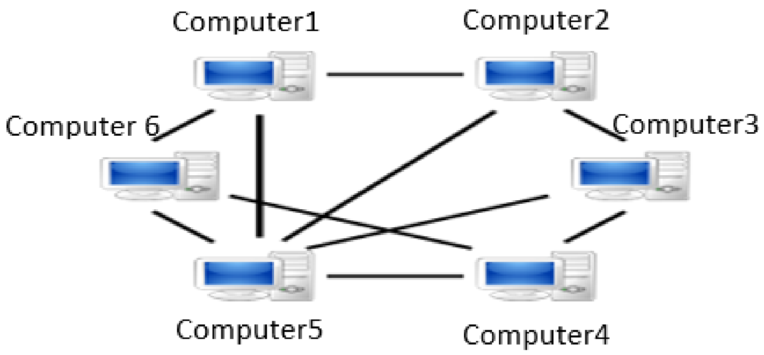

Figure (1): P2P (Peer-to-Peer) distributed systems (Anton and Norbert 2016).

b. Clusters Distributed Systems

A computer group is a group of interconnected computers that are habitually linked using a fast local area network (LAN). Computer clusters are the preferred distributed system construction type because of the top performance-price divided computing ratio. It makes up the rest are Massively Parallel Processors (MPP) and Pleiades systems (Grudenić and Bogunović 2009). It's a local area network with high-speed connections and more common in modern high-performance computing.

Number of computers are grouped so that benefiting from an individual resource group. Then, the result of the minimum job will be merged to form the end outcome (Amir et al. 2004). Cluster computing aid organizations raise their computing power by apply generally available standard technology. These hardware and software, known as commodities, can be bought on the market at a comparatively small cost(Lakshmanan, Ahamad, and Venkateswaran 2003). A set of computers combination together in such a way that a single resource group makes a group. Then, the result of the minimum job will be merged to form the end outcome(Amir et al. 2004).

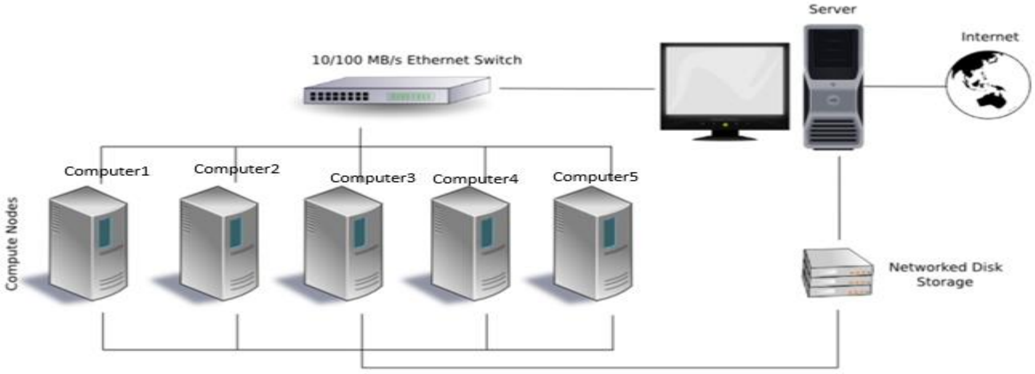

Figure (2): Clusters distributed systems (Sweilam, Moharram, and Ahmed 2012).

c. Grids Distributed Systems

Grid computing is a group of computational resources from a variety of administrative areas used for a usual task, technical, or commercial problem that requires a large number of computer training courses or the need for large amounts of data processing. It is a kind of parallel distributed system that allows you to dynamically distribute, select, and edit independent geographically distributed sources over time, depending on the availability, power, performance, cost, and quality of service requirements for users (Saafan 2009)(Rashid et al. 2018).

The broker used in Network Transactions is responsible for assigning tasks. The size of a transition system can vary from a few hundred computers in a large organization to thousands of nodes in many organizations. Smaller networks that are confined to one organization are commonly known as a company within a node, while a larger system is referred to as a joint node (Puttaswamy, Zheng, and Zhao 2008).

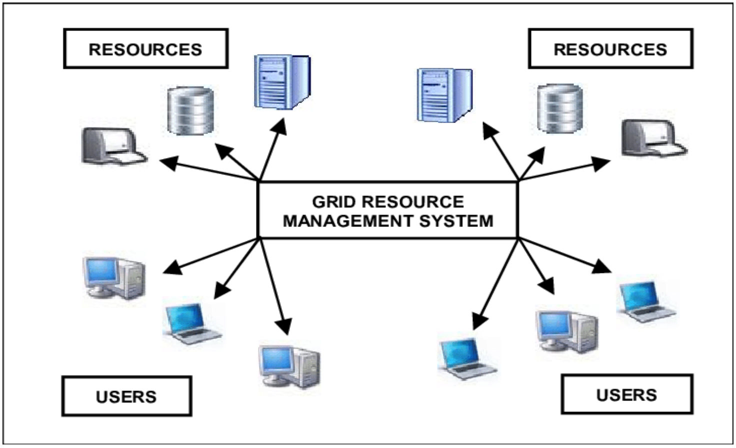

Figure (3): Grids distributed systems (Kaur, Kaur, and Kaur 2013).

d. Distributed Storage Systems

The rapid growth of storage capacity, account resources, and bandwidth combined with the lower cost of storage equipment has increased the popularity of distributed storage systems(Harinath et al. 2015). The main kind of distributed storage systems. In particular, redundant array of inexpensive disks (RAID), central raid, network storage, and Local Area network (LAN). Network-attached storage NAS and storage area network SAN are the four most common distributed storage technologies.

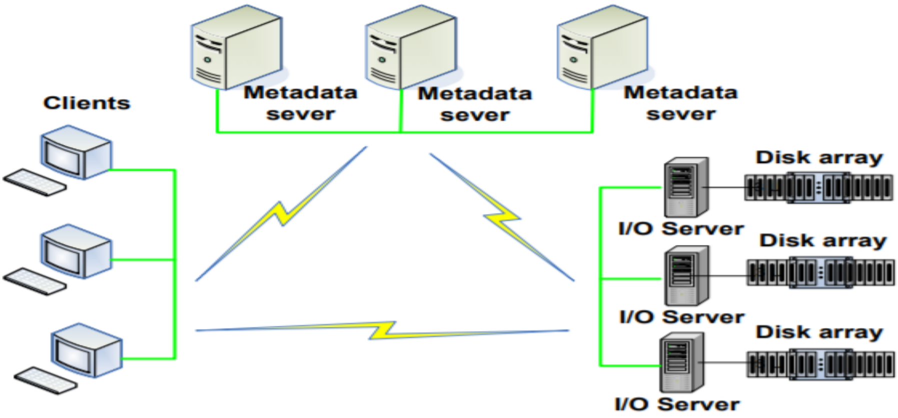

Figure (4): Distributed storage systems (Chen et al. 2011).

3. Multicore

Usually, a multi-core processor is a single processor that has several cores on a chip(Wang et al. 2010). This is due to the presence of parallel processing technology that was absent in a single-core processor. There are CPUs with different basic levels two-cores, four-cores, six-cores, eight cores, ten cores, and more (Sondhi and Ganesh n.d.) (Zeebaree et al. 2020). The multi-core processor is an integrated circuit (IC) with two or more processors linked to improve performance, reduce energy consumption, and handle multiple tasks more efficiently and simultaneously (Rouse 2013).

The multi-core processor can be classified into two types: homogeneous and heterogeneous. Symmetric multiprocessor (SMP) operating systems are usually implemented on monolithic multi-core processors to compute high-performance combinations. On the other hand, heterogeneous multi-core processors, which consist of various cores dedicated to specific applications, are better for embedded systems(Wei et al. 2011).

In recent years, CPU designers have moved from a multi-core architecture with multiple processing threads to a clock speed of multi-core processors (Rao, Prasad, and Venkateswarlu 2009). The core can be considered as one healer. The dual-core contains two internal processors, which are produced by a single chip. The quad-core model has four processors, two dual-processors, made on one chip. More cores are useful for a variety of tasks. Single-thread applications can use only one kernel, leaving any other procedure idle. Core i3 processors have four cores, and i5 and i7 has four cores (Rao, Moturi, and KLEF 2018).

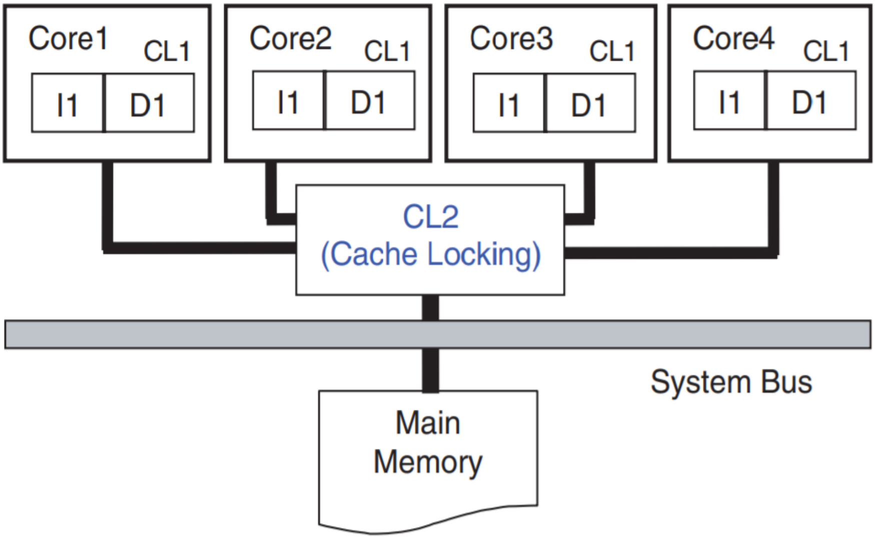

Figure (5): multi-core distributed system (Asaduzzaman and Gunasekara 2012).

4. Literature Review

V. Sklyarov, et. al. (Sklyarov et al. 2016) in 2016, explored distributed computing systems that can be used efficiently to process frequently requested information in electronic, environmental, medical and biological applications. Pre-processing can be done in very parallel accelerators that are set to reconfigurable devices. The core of the accelerator is a sort/search network that is implemented either in FPGA or in a programmable chip-based system (such as ZYNQ devices). Data is sent to the computer via the high-bandwidth PCI-Express bus.

B. Sreevidya, et. al. (Sreevidya, Rajesh, and Mamatha 2018) in 2018, worked on determination of sensor network performance after on-demand accuracy using an RSA-based security scheme that allows fake data to enter the wireless sensor network(WSN) using NS2 emulation. The RSA algorithm applied in the modified scheme provides large system security than the MAC algorithm of the BECAN scheme. The MAC algorithm only denoted system authentication, but the RSA algorithm constructed the system more security with public key encryption technology.

S. Phoemphon, et. al. (Phoemphon, So-In, and Nguyen 2018) in 2018, studied the possible integration of two computing technologies, fuzzy logic (FL) and extreme learning machines (ELMs), with the aim of improving the estimated localization accuracy, taking into account the above factors. Localization is one of the major challenges encountered in wireless sensor networks, especially in the non-attendance of such GPS installation equipment. Unlike ELMs, FL methods provide high accuracy under limited node density and coverage conditions.

M. Hossain, et. al. (Hossain et al. 2018) in 2018, addressed the designed network measured latency and throughput compared to the interactive real-time online (ROIA) multi-box schedule and NOX. The main technology of SDN application is data level and management level separation, and network virtualization through programming. The total amount of time a user can respond is called the response time. The transfer rate is the exchange rate at which the network transmits data.

A. Rauniyar, et. al. (Rauniyar, Engelstad, and Moen 2018) in 2018, proposed a new distributed localization algorithm is using SL-PSO based social learning for the Internet of Things. With the SLPSO algorithm, the method aims to precisely localize diffuse sensor nodes and reduce computational complexity that will further enhance the lifespan of these sensing nodes with limited resources. Extensive simulations are performed to demonstrate the effective performance of the SL-PSO algorithm in fine localization.

Y. Liu, et. al. (Liu et al. 2019) in 2019, proposed a lightweight block-chain system called a lightweight chain that is resource-efficient and suitable for energy Industrial Internet of Things (IIoT) scenarios. In particular, a green agreement mechanism called symmetric multicore processer (SMP) to facilitate collaboration among IIoT devices, a light data structure called light-block (LB) to identify broadcast content. Moreover, they designed a new block-chain to avoid unlimited overhead growth without affecting the block-chain backlog.

F. Al-Wesabi, et. al. (Al-Wesabi, Iskandar, and Ghilan 2019) in 2019, expanded the work of the E-Avala approximation to improving the overall performance of the work-study by increasing power recovery capabilities. The work-study was executed and compared with the previous study using square modelling with different configuration parameters. The comparative results show that the proposed study has allowed for better improvement with recycling capabilities at different levels.

L. Tan, et. al. (Tan, Zhao, and Zhang 2019) in 2019, demonstrated using iTrace to collect effects on iOS. This creates opportunities for many types of searches that were previously impossible to perform on iOS. The iTrace Ultra Speed also allows for time-sensitive analysis. The collected system calls for effects for commonly used applications and summarizes and displays their results from these effects. They hoped to highlight application behaviour and open up areas for further investigation.

F. Noor, et. al. (Noor, Ibrahim, and AlKhattab 2020) in 2020, proposed the parallel distributed bat algorithm (PDBA) using the Message Passing Interface (MPI) of the computer group in the programming C language. PDBA time and complexity are determined. Also, the results were presented in terms of speed, efficiency, time completed number of times to perform the fitness function. An algorithm is an optimization algorithm that is workable and efficient to obtain the best approximate solutions to nonlinear problems.

M. Bhatia, et. al. (Bhatia, Sood, and Kaur 2020) in 2020, proposed a quantitative approach to the planning of homogeneous functions in fog-based applications. In particular, the scale of a given node is determined by calculating the node computational index to estimate the computational power of the fog computing nodes. Moreover, the QCI neural network model is proposed to optimize the node to handle the real-time mock task. Comparative analysis was performed using advanced scheduling models such as heterogeneous completion time, Min-Max, and Round Robin for comparative analysis to determine performance improvements.

K. Warasup, et. al. (Warasup, Hamamura, and Pattaramalai 2020) in 2020, introduced a new MAC protocol that enhances network performance for wireless LAN with Multiple

Packet Reception (MPR) capability. The main feature of their proposed protocol is that multiple asynchronous RTS transmissions are allowed. As a result, the method get a higher probability of packet transmission when the network is operating at the maximum MPR capacity. Moreover, they are developed a mathematical model to evaluate the proposed protocol.

D. K. Kotary, et. al. (Kotary and Nanda 2020) in 2020, compared with Diffusion Particle Swarm Optimization (DPSO), Diffusion Whale Optimization Agorithm (DWOA), Dffusion Elephant Herding Optimization (DEHO), Diffused K-Mans (DK-Mans) Dunn Index, and Outline Index In terms of time complexity. Performed on two real groups of artificial data sets. The minimum average Euclidean deviation (EAD) for improving farm butterfly flame for agricultural farm data proportion compared to DPSO, DWOA, DEHO, and DK-Mans is 7.16%, 3.25%, 5.24%, and 21.70 %.

Bianco, et. al. (Bianco 2020) in 2020, discussed the design and development of FPGA-based NIC enabling us to overcome performance barriers bottlenecks and Lack of flexibility in commercial NICs. Network interface cards (NICs) are gaining more and more attention in the research community as they can provide several gigabytes of transmission speeds per packet, a performance similar to low-to-mid-range routers. The effectiveness of the proposed approach and limits was comprehensively discussed.

P. Neelima, et. al. (Neelima and Reddy 2020) in 2020, introduced the ADA-based, multi-load balancing method, and this system is being tested through simulation experiments. They have planned an approach of the design to deliver a well-balanced load on virtual machines deliver them on time at the lowest cost. Moreover, the proposed algorithm demonstrates the ADA's ability to improve task scheduling and resource allocation in the cloud computing environment. To assess the performance of the proposed system, three problematic cases are analysed.

S. Yousefi, et. al. (Yousefi et al. 2020) in 2020, introduced a new route planning mechanism for collecting data on the Internet. The proposed mechanism includes two basic states. Gather data on the Internet of Things using the idea of mobile software agents. The first step is to integrate IoT devices. The Group leaders assembled groups by assigning mobile agents to a wedge-based operation. The basic purpose of the second period is to provide a path mapping to each mobile factor of group leaders to effectively collect data through the Markov Determination Process (MDP).

F. Shahid, et. al. (Shahid, Khan, and Jeon 2020) in 2020, proposed Distributed Ledger for Internet of Things (DL-for-IoT) addresses the challenge of integrating the two technologies: distributed book and Internet of things. Notable features of DL-for-IoT include transaction hierarchy, floor cuts, lightweight compatibility, and quantitatively secure digital signatures. The basic building block of DL-for-IoT is a new one-time signature (OTS) method called DL-OTS. By comparing DL-OTS with general one-time signature schemes, they concluded that DL-OTS is a compact and high-speed energy-saving signature scheme.

X. Zhao, et. al. (Zhao et al. 2020) in 2020, designed and developed based on distributed heterogeneous medical data a system to share and merge the distributed medical data and apply it to some internal hospitals. Therefore, a series of smart medical information network platform system is created so that patients can enjoy the service. High quality, safe, and proper diagnosis and dealing are based on little waiting times and vital medical costs. Finally, the results of questionnaires and interviews concluded that they were compared to previous treatments.

S. Dhingra, et. al. (Dhingra et al. 2020) in 2020, introduced a cloud fog for real-time analysis of the network of basic cloud services for traffic monitoring base to overcome the clogging latency limits. Therefore, proposed to implement a prototype Smart Traffic Monitoring System (STMS) and it is designed for signal traffic monitoring of overcrowding. It can also be adapted to detect road accidents that require immediate assistance in case of congestion. Within this framework, the small computer on the module acts as a fog node, collecting real-time data from geographically dispersed sensors, moving it to the cloud for storage and processing.

Parra, et. al. (Parra et al. 2020) in 2020, proposed cloud distributed deep learning framework for detecting the phishing attacks. The model consists of two main security mechanisms that operate cooperatively: first, a distributed convolutional neural network (DCNN) model included as an add-on partial security for the Internet of Things device to detect phishing and denial of distributed service application (DDoS). Second, a cloud-based Long-Short Term Memory (LSTM) model hosted on the backside to detect Botnet attacks and CNN embedding to detect phishing attacks distributed across many IoT devices. Distributed CNN model, included in the ML engine of customer's IoT device.

Y. Gao, et. al. (Gao et al. 2020) in 2020, evaluated and compared FL and SplitNN IoT settings in the real world in terms of learning performance and overall hardware application. considered a wide variety of data sets, different model structures, multiple clients, and different performance metrics. For learning performance, determined by model accuracy and convergence velocity measures, evaluated the empirically for both FL and SplitNN under different types of data distributions such as unbalanced, non-independent, and identically distributed data (other than IID)

Y. Ren, et. al. (Ren et al. 2020) in 2020, introduced the innovative DCOMB (Dual Combination Bloom Filter) method to convert the processing power of bitcoin mining into query processing power. In addition, the use of the DCOMB method to create a block chain based IoT data query model. This model combines the flow of IoT data with a block chain of the timing chain, improving the interoperability and versatility of the data of the IoT database system.

W. Yánez, et. al. (Yánez et al. 2020) in 2020, proposed a new context-sensitive mechanism for distributing chain data across IoT blockchain systems. In particular, they developed the data controller based on tasteless logic to calculate the cost allocation value for each data request, taking into account multiple context parameter data network quality and their chain distribution. Moreover, introduced how the design and perception of improves the architectural styles of the two common uses of the Internet of Things (block, fog).

5. Discussion

This part depends on the details explained in table 1. All the above literature used different techniques in distributed systems and the significant result. For example. The first literature V. Sklyarov, et al. the Scheduling algorithms used and the results of pre-processing, statistical treatment, analysis of existing, acquired groups and data mining. Also, B. Sreevidya, et al. the MAC and RSA algorithm used and the result indicate that schema performance is better than the current schemes that provide either data integrity or sender authentication. Furthermore, S. Phoemphon, et al. the ELM algorithm used and the result derivatives and improvements, Fuzzy Centroid and ELMs.

In addition, each piece of literature uses a different operating system and different versions. For example L. Tan, et al. used iOS 4.1 and Android 2.3.6, G. D. L. T. Parra used Linux (Ubuntu 16.04) and Y. Ren, et al. used windows 10. On the other hand, the programming language is the crucial part of any paper, and all the above literature uses different programming languages, such as Matlab, Java, Python, C ++. Also, the network is the tower part for the distributed system because the literature shows that if they use a good network and fast internet, they will get a good outcome. Finally, all the above literature used an excellent technique and found a significant result: one is the best and gets high performance for system displayed at the last part of this paper.

Table (1): Analysis of surveyed researches

Literature	Technique	Significant Results	Operating System	CPU type	Programing	Networking approach
(Sklyarov et al. 2016)	Scheduling algorithms	The results of pre-processing, statistical treatment, analysis of existing, acquired groups and data mining.	32-bit Linux	dual-core	C language	Wired
(Sreevidya et al. 2018)	MAC and RSA algorithm	Results indicate that the schema performance is better than the current schemes that provide either data integrity or sender authentication.	Linux	-	C++ language	Wired and wireless
(Phoemphon et al. 2018)	ELM algorithm	Traditional free domain localization plans consisting of Centroid, derivatives and improvements, Fuzzy Centroid and ELMs.	Windows	Quad Core (TM) Q8400	MATLAB language	Wireless
(Hossain et al. 2018)	SDN and QOS technique	Easily configure traffic without touching any individual keys and services needed in the network.	Linux	-	PYTHON language	Wired
(Rauniyar et al. 2018)	SL-PSO algorithm	Significantly reduced mean localization error compared to the optimization of conventional particle swarm (PSO).	Windows 7 64-bit	Intel Core i7-6500	MATLAB R2016b	Wireless

(Liu et al. 2019)	Light Block and UBOF algorithm	The individual computational cost can reduce to 39.32% and speed up the block generation by up to 74.06%.	GUN/Linux 4.8.0-36	Intel Xeon 2.5	Python	Wireless
(Al-Wesabi et al. 2019)	E-Avala algorithm	It gave a better performance with different levels of processing capabilities.	Windows 7	Intel Core™i5 M450/2.40	JAVA language	-
(Tan et al. 2019)	iTrace	The difference in the frameworks used or developed using a different (logical) application to achieve the same function.	iOS 4.1 and Android 2.3.6	Cortex A8, 1 GHz	C language	Mobile(wire less)
(Noor et al. 2020)	Bat algorithm	Effective for optimization problems with large population size and extensive search space	Linux	-	C language	Wireless
(Bhatia et al. 2020)	QCI-load scheduling algorithm	The presented approach is extremely effective in optimizing the contract computing of fog in wireless applications in real-time	-	Intel Core TM i7-4710MQ	JAVA language	Wireless
(Warasup et al. 2020)	MU-MIMO	High bandwidth transmission potential when network is running at full MPR capacity	Windows	-	MATLAB language	Wireless
(Kotary and Nanda 2020)	DMFO algorithm	Increasing sensor data size or sensor node number.	Windows 8 (64) bit	Core i3	MATLAB language	Wireless

(Bianco 2020)	scheduling algorithm	Processed and managed in the software by the operating system, thereby reducing charging performance.	Linux	PCI-X core	LOGIC design	Wireless packet network (WPN)
(Neelima and Reddy 2020)	Adaptive Dragonfly algorithm (ADA)	A better approach achieves the result of load balancing compared to other approaches.	Windows 7 (64)bit	dual-core	Java	-
(Yousefi et al. 2020)	k-medoids algorithm	The proposed mechanism improves power consumption, data transmission delay, and IoT reliability	Windows 7	Intel (R) Core i7-3520 M	-	Wireless
(Shahid et al. 2020)	DL-OTS algorithm	76% reduction at the time of signature and 48.7% energy savings compared to Winter-Nitz-OTS scheme.	Windows 8.1 32-bit	Intel Core i5	Python	Wireless
(Zhao et al. 2020)	AES algorithm	The participation of smart medicine among the general public increased by about 79%	Linux (Ubuntu 12.04) 32 bit	Intel Q9300	Python	Wireless
(Dhingra et al. 2020)	STMS technique	Improved cloud platform performance in terms of reduced response time and increased bandwidth.	Windows (64) bit	dual-core	(Arduino IDE) Java	Wireless

(Parra et al. 2020)	DCNN and LSTM model	Detection results for specific attacks when the model is trained with benign and malicious data	Linux (Ubuntu 16.04)	Intel Xeon e5-2670 V3	Python	Wireless
(Gao et al. 2020)	FL and Split-NN model	Our experimental results of learning performance at Split-NN research questions for future investigations	windows 10	Intel core i7-7700HQ	Python	Wireless
(Ren et al. 2020)	DCOMB model	The results showed that the random read performance of DCOMB query is higher than COMB	windows 10	Intel i3	Java	-
(Yánez et al. 2020)	IoT-blockchain	Experimental results indicate that realizing a data allocation mechanism improves network latency, blockchain storage and reduces power consumption.	Windows 7	Intel Core 2 Duo	Java and MySQL	Wireless

6. Conclusion

Distributed systems are designed separately from the core network, making the worst assumptions. The multicore processor can be classified into two types: homogeneous and heterogeneous. This paper reviewed of Performance Impact of Distributed-Memory Parallel Processing Approach on Performance Enhancing of Multicomputer-Multicore Systems. Also, number of methods been introduced which used Impact of Distributed-Memory Parallel Processing Approach on Performance Enhancing of Multicomputer-Multicore Systems. Adding to that, the best methods been explained, with focusing on speed for performance enhancing of multicore in the distributed systems.

Depending on the details explained in Table I in the discussion section, it can be concluded that the best methods were those which depended by Y. Liu et al. and F. Shahid, et al. Firstly, Liu et al. used an operating system named gun/Linux 4.8.0-36, intel Xeon 2.5, python programming language. The outcome of the proposed method made a time reduction by 39.32% and speeded up the block generation up to 74.06% of individual computational cost. Secondly, F. Shahid et al. used an operating system named Windows 8.1 32-bit Intel core i5. The programming language is python, and result, it can reduce 76% at the time of signature and 48.7% energy savings compared to winternitz-ots scheme.

References:

Agarwal, V. 2004. “Fault Tolerance in Distributed Systems, I. Institute of Technology Kanpur.”

Al-Wesabi, Fahd, Huda Iskandar, and Mokhtar Ghilan. 2019. “Improving Performance in Component Based Distributed Systems.” *EAI Endorsed Transactions on Scalable Information Systems* 6(22).

Amir, Yair, Yongdae Kim, Cristina Nita-Rotaru, John L. Schultz, Jonathan Stanton, and Gene Tsudik. 2004. “Secure Group Communication Using Robust Contributory Key Agreement.” *IEEE Transactions on Parallel and Distributed Systems* 15(5):468–80.

Anton, B., and A. Norbert. 2016. “Peer to Peer System Deployment.” *Acta Electrotech. Inform* 16:11–14.

Asaduzzaman, Abu, and Govipalagodage H. Gunasekara. 2012. “A Way Cache Locking Scheme Supported by Knowledge Based Smart Preload Effective for Low-Power Multicore Electronics.” *Journal of Low Power Electronics* 8(5):552–64.

Bansal, Sanjay, Sanjeev Sharma, and Ishita Trivedi. 2011. “A Detailed Review of Fault-Tolerance Techniques in Distributed System.” *International Journal on Internet & Distributed Computing Systems* 1(1).

Bhatia, Munish, Sandeep K. Sood, and Simranpreet Kaur. 2020. “Quantumized Approach of Load Scheduling in Fog Computing Environment for IoT Applications.” *Computing* 1–19.

Bianco, Michele Petracca Robert Birke Andrea. 2020. “HERO: High-Speed Enhanced Routing Operation in Ethernet NICs for Software Routers⋆.”

Blake, Geoffrey, Ronald G. Dreslinski, and Trevor Mudge. 2009. “A Survey of Multicore Processors.” *IEEE Signal Processing Magazine* 26(6):26–37.

Bridges, Matthew J., Neil Vachharajani, Yun Zhang, Thomas Jablin, and David I. August. 2008. “Revisiting the Sequential Programming Model for the Multicore Era.” *IEEE Micro* 28(1):12–20.

Chen, Xin, Xubin He, He Guo, and Yuxin Wang. 2011. “Design and Evaluation of an Online Anomaly Detector for Distributed Storage Systems.” *J. Softw.* 6(12):2379–90.

Dhingra, Swati, Rajasekhara Babu Madda, Rizwan Patan, Pengcheng Jiao, Kaveh Barri, and Amir H. Alavi. 2020. “Internet of Things-Based Fog and Cloud Computing Technology for Smart Traffic Monitoring.” *Internet of Things* 100175.

Dino, Hivi Ismat, Subhi R. M. Zeebaree, Omar M. Ahmad, Hanan M. Shukur, Rizgar R. Zebari, and Lailan M. Haji. n.d. “Impact of Load Sharing on Performance of Distributed Systems Computations.”

Fiat, Amos, and Jared Saia. 2002. “Censorship Resistant Peer-to-Peer Content Addressable Networks.” Pp. 94–103 in *SODA*. Vol. 2.

Gao, Yansong, Minki Kim, Sharif Abuadbba, Yeonjae Kim, Chandra Thapa, Kyuyeon Kim, Seyit A. Camtepe, Hyoungshick Kim, and Surya Nepal. 2020. “End-to-End Evaluation of Federated Learning and Split Learning for Internet of Things.” *ArXiv Preprint ArXiv:2003.13376*.

Grudenić, Igor, and Nikola Bogunović. 2009. “Computer Cluster and Grid Simulator.” in *MIPRO*

2009.

Haji, Lailan M., Subhi R. M. Zeebaree, Omar M. Ahmed, Amira B. Sallow, Karwan Jacksi, and Rizgar R. Zeabri. 2020. “Dynamic Resource Allocation for Distributed Systems and Cloud Computing.”

Harinath, Depavath, K. Ramesh Babu, P. Satyanarayana, and M. V. Ramana Murthy. 2015. “Enhancing Security Using Video Steganography and WaterMarking.” *European Journal of Applied Sciences* 3(5):1.

Hossain, Md, Mohammad Nowsin Amin Sheikh, Shawon Rahman, Sujan Biswas, and Md Ariful Islam Arman. 2018. “Enhancing and Measuring the Performance in Software Defined Networking.” *International Journal of Computer Networks & Communications (IJCNC)* 10(5).

Kaur, Ramanjyot, Tajinder Kaur, and Harpreet Kaur. 2013. “Scheduling in Grid Computing Environment.” *International Journal of Advanced Research in Computer Science and Software Engineering* 3(6):455–58.

Kotary, Dinesh Kumar, and Satyasai Jagannath Nanda. 2020. “Distributed Robust Data Clustering in Wireless Sensor Networks Using Diffusion Moth Flame Optimization.” *Engineering Applications of Artificial Intelligence* 87:103342.

Lakshmanan, Subramanian, Mustaque Ahamad, and H. Venkateswaran. 2003. “Responsive Security for Stored Data.” *IEEE Transactions on Parallel and Distributed Systems* 14(9):818–28.

Liu, Yinqiu, Kun Wang, Yun Lin, and Wenyao Xu. 2019. “$\mathsf {LightChain} $: A Lightweight Blockchain System for Industrial Internet of Things.” *IEEE Transactions on Industrial Informatics* 15(6):3571–81.

Lu, Gang, Jianfeng Zhan, Xinlong Lin, Chongkang Tan, and Lei Wang. 2016. “On Horizontal Decomposition of the Operating System.” *ArXiv Preprint ArXiv:1604.01378*.

McCool, Michael D. 2008. “Scalable Programming Models for Massively Multicore Processors.” *Proceedings of the IEEE* 96(5):816–31.

Neelima, P., and A. Rama Mohan Reddy. 2020. “An Efficient Load Balancing System Using Adaptive Dragonfly Algorithm in Cloud Computing.” *Cluster Computing* 1–9.

Noor, Fazal, Abdulghani Ibrahim, and Mohammed M. AlKhattab. 2020. “Performance of

Parallel Distributed Bat Algorithm Using MPI on a PC Cluster.” *Annals of Emerging Technologies in Computing (AETiC)* 4(1).

Parra, Gonzalo De La Torre, Paul Rad, Kim-Kwang Raymond Choo, and Nicole Beebe. 2020. “Detecting Internet of Things Attacks Using Distributed Deep Learning.” *Journal of Network and Computer Applications* 102662.

Phoemphon, Songyut, Chakchai So-In, and Tri Gia Nguyen. 2018. “An Enhanced Wireless Sensor Network Localization Scheme for Radio Irregularity Models Using Hybrid Fuzzy Deep Extreme Learning Machines.” *Wireless Networks* 24(3):799–819.

Ports, Dan R. K., Jialin Li, Vincent Liu, Naveen Kr Sharma, and Arvind Krishnamurthy. 2015. “Designing Distributed Systems Using Approximate Synchrony in Data Center Networks.” Pp. 43–57 in *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*.

Puttaswamy, Krishna P. N., Heather Zheng, and Ben Y. Zhao. 2008. “Securing Structured Overlays against Identity Attacks.” *IEEE Transactions on Parallel and Distributed Systems* 20(10):1487–98.

Ramanathan, R. M. 2006. “Intel Multi-Core Processors: Making the Move to Quad-Core and Beyond.” *Technology@ Intel Magazine* 4(9):2–4.

Rao, S. N. Tirumala, Sireesha Moturi, and Vaddeswaram KLEF. 2018. “SCALABILITY OF MEMORY MAPPED K-MEANS ALGORITHM WITH HYPER-THREADED PROCESSORS.” *International Journal of Pure and Applied Mathematics* 120(6):4179–92.

Rao, S. N. Tirumala, E. V Prasad, and N. B. Venkateswarlu. 2009. “A Scalable K-Means Clustering Algorithm on Multi-Core Architecture.” Pp. 1–9 in *2009 Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS)*. IEEE.

Rashid, Zryan Najat, Subhi R. M. Zebari, Karzan Hussein Sharif, and Karwan Jacksi. 2018. “Distributed Cloud Computing and Distributed Parallel Computing: A Review.” Pp. 167–72 in *2018 International Conference on Advanced Science and Engineering (ICOASE)*. IEEE.

Ratnasamy, Sylvia, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. 2001. “A Scalable Content-Addressable Network.” Pp. 161–72 in *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*.

Rauniyar, Ashish, Paal Engelstad, and Jonas Moen. 2018. "A New Distributed Localization Algorithm Using Social Learning Based Particle Swarm Optimization for Internet of Things." Pp. 1–7 in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE.

Ren, Yongjun, Fujian Zhu, Pradip Kumar Sharma, Tian Wang, Jin Wang, Osama Alfarraj, and Amr Tolba. 2020. "Data Query Mechanism Based on Hash Computing Power of Blockchain in Internet of Things." *Sensors* 20(1):207.

Rouse, Margaret. 2013. "Definition: Multi-Core Processor." *TechTarget. Retrieved March* 6.

Rowstron, Antony, and Peter Druschel. 2001. "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems." Pp. 329–50 in *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer.

Saafan, Ahmed. 2009. "Distributed Denial of Service Attacks: Explanation, Classification and Suggested Solutions." *23 March 2009)[Online]. Available: Http://Www. Exploitdb. Com*.

Shahid, Furqan, Abid Khan, and Gwanggil Jeon. 2020. "Post-Quantum Distributed Ledger for Internet of Things." *Computers & Electrical Engineering* 83:106581.

Shukur, Hanan, Subhi Zeebaree, Rizgar Zebari, Omar Ahmed, Lailan Haji, and Dildar Abdulqader. 2020. "Cache Coherence Protocols in Distributed Systems." *Journal of Applied Science and Technology Trends* 1(3):92–97.

Sklyarov, V., Artjom Rjabov, Iouliia Skliarova, and Alexander Sudnitson. 2016. "High-Performance Information Processing in Distributed Computing Systems." *International Journal of Innovative Computing, Information and Control* 12(1):139–60.

Sondhi, Garima, and Abhinaya Ganesh. n.d. "An Overview of Multi Core Processors."

Sreevidya, B., M. Rajesh, and T. M. Mamatha. 2018. "Design and Development of an Enhanced Security Scheme Using RSA for Preventing False Data Injection in Wireless Sensor Networks." Pp. 225–36 in *Ambient Communications and Computer Systems*. Springer.

Van Steen, Maarten, and Andrew S. Tanenbaum. 2017. *Distributed Systems*. Maarten van Steen Leiden, The Netherlands.

Stoica, Ion, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. 2001. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications." *ACM*

SIGCOMM Computer Communication Review 31(4):149–60.

Sweilam, N. H., H. M. Moharram, and Sameh Ahmed. 2012. “On the Parallel Iterative Finite Difference Algorithm for 2-D Poisson’s Equation with MPI Cluster.” Pp. MM–78 in *2012 8th International Conference on Informatics and Systems (INFOS)*. IEEE.

Tan, Lucas, Fuyao Zhao, and Xu Zhang. 2019. “15712 Advanced Operating and Distributed System Android and IOS Platform Study Final Report.”

Wang, Chonggang, and Bo Li. 2003. “Peer-to-Peer Overlay Networks: A Survey.” *Department of Computer Science, The Hong Kong University of Science and Technology, Hong Kong* 9.

Wang, Lizhe, Jie Tao, Gregor von Laszewski, and Holger Marten. 2010. “Multicores in Cloud Computing: Research Challenges for Applications.” *JCP* 5(6):958–64.

Warasup, Kittipong, Masanori Hamamura, and Suwat Pattaramalai. 2020. “Performance Analysis for Asynchronous Requests in Wireless Networks with Multiple Packet Reception Capability.” *Wireless Personal Communications* 1–23.

Wei, Ting-Ying, Zhi-liang Qiu, Chung-ping Young, and Da-Wei Chang. 2011. “Development of Heterogeneous Multi-Core Embedded Platform for Automotive Applications.” in *2011 International Conference on Circuits, System and Simulation (IPCSIT)*. Vol. 7.

Yánez, Wendy, Redowan Mahmud, Rami Bahsoon, Yuqun Zhang, and Rajkumar Buyya. 2020. “Data Allocation Mechanism for Internet-of-Things Systems With Blockchain.” *IEEE Internet of Things Journal* 7(4):3509–22.

Yousefi, Shamim, Farnaz Derakhshan, Hadis Karimipour, and Hadi S. Aghdasi. 2020. “An Efficient Route Planning Model for Mobile Agents on the Internet of Things Using Markov Decision Process.” *Ad Hoc Networks* 98:102053.

Zeebaree, Subhi R. M., Lailan M. Haji, Imad Rashid, Rizgar R. Zebari, Omar M. Ahmed, Karwan Jacksi, and Hanan M. Shukur. 2020. “Multicomputer Multicore System Influence on Maximum Multi-Processes Execution Time.”

Zhao, Ben Yanbin, John Kubiatowicz, and Anthony D. Joseph. 2001. “Tapestry: An Infrastructure for Fault-Tolerant Wide-Area Location and Routing.”

Zhao, Xin, Wei Xiao, Lu Wu, Zhigang Zhao, Jidong Huo, Shi Wang, Zhenhua Guo, and Dianmin Sun. 2020. “Intelligent City Intelligent Medical Sharing Technology Based on Internet of

Things Technology.” *Future Generation Computer Systems*.

کاریگەری چارەسەرکردنی بەلاوکردنەوەی میموری بۆ باشترکردنی بەرێوەچونی ژمارەیەك کومبیوتەر و ژمارەیەك پروسێسەرەکان: پێداچونەوە

کورتکراوە: بەلاوکردنەوەی میموری بەکاردێت بو زانستەکان کومبیوتەر لەسەر شێوازی سیستەمی کومیبوتەر و ژمارەیەك پروسیسەر بە هەموو چارەسەرکردن و تایبەتەکانی میموری. دەتوانێت بەکاربێهنرێت بۆ کارەکانی ژماردن و داتایە ناوخویەکان بە تەنیا, لە بەرئەوە ئەگەر پێویستیەکان زوربون دەتوانێت بەکاربێهترێت بو هەندێك داتاکانی تر. دەبێت پەیوەندی بکەی بەیەکێك یان زورتر لە کومبیەتەرەکان و پروسێسەرەکانی دور بو بەلاو دردنەوەی داتاکانی کومبیوتەر. هەروەها بەلاوکردنەوەی کومبیوتەر یەکێکە لە ئامێرەکانی چارەسەرکردنی ژمارەکان بەیەکسانی و چارەسەرکردنی زورێك لە کێشەکان بو جێ بە جێ کردنی ژمارەیك پروسێسەر. بەلاوکردنەوەی سیستەمەکان بە شێوەیەکی جیاکراو دیزایین کراوە لە تورەکانی سەرەکی. زور جۆری جیاواز هەیە بو بەلاوکردنەوی سیستەمەکان بو نمونە: تورێن پیر تو پیر(p2p) , گروپەکان , تورەکان , هەلگرتن و بەلاوکردنەوەی سیستەمەکان. ژمارەی کوی پروسێسەرەکان دابەش دەکرێن بۆ دوو جورا:گونجاندنی و ناگونجاندنی پروسێسەرەکان.ئەم نامیلکەیە پێداچونەوەیەکە لسەر کاریگەری چارەسەرکردنی بەلاوکردنەوەی میموری بۆ باشترکردنی بەرێوەچونی ژمارەیەك کومبیوتەر و ژمارەیەك پروسێسیەر. هەروەها بو پێشاندان و رونکردنەوەی ژمارەیەك بابەت لە بکار هێنانی بەلاوکردنەوەی میموری سیستەم ودیاریکردنی باشترین شیکەروە لسەر ژماربەك لە چالاکی بەلاوەکردنەوەی سیستەمەکانی میموری.

تأثير نهج المعالجة المتوازية للذاكرة الموزعة على تحسين أداء الأنظمة متعددة الحواسيب ومتعددة النوى: مراجعة

الملخص:

الذاكرة الموزعة هي مصطلح يستخدم في علوم الكمبيوتر لوصف نظام كمبيوتر متعدد المعالجات يكون لكل معالج ذاكرة خاصة به. يمكن أن تعمل الوظائف الحسابية مع البيانات المحلية فقط، لذلك إذا كنت بحاجة إلى بيانات عن بُعد، فسيتعين عليك الاتصال بواحد أو أكثر من المعالجات البعيدة. كثيرًا ما تُستخدم الحوسبة المتوازية والموزعة معًا. تستخدم الحوسبة المتوازية الموزعة العديد من أجهزة الحوسبة لمعالجة المهام

بالتوازي، بينما تستخدم الحوسبة المتوازية على جهاز كمبيوتر واحد معالجات متعددة لتنفيذ المهام بالتوازي. تم تصميم الأنظمة الموزعة بشكل منفصل عن الشبكة الأساسية. هناك أنواع مختلفة من الأنظمة الموزعة مثل شبكات الند للند (P2P) والمجموعات والشبكات وأنظمة التخزين الموزعة. يمكن تصنيف المعالج متعدد النواة إلى نوعين: متجانس وغير متجانس. تستعرض هذه الورقة تأثير نهج المعالجة المتوازية للذاكرة الموزعة على تحسين أداء الأنظمة متعددة الحواسيب ومتعددة النواة. نقدم أيضًا عددًا من الأساليب المستخدمة في أنظمة الذاكرة الموزعة ونناقش الطريقة الأفضل وتعزز الأداء متعدد النواة في الأنظمة الموزعة.