

## A Survey on Parallel Multicore Computing: Performance & Improvement

Ola Surakhi\*, Mohammad Khanafseh, Sami Sarhan

University of Jordan, King Abdullah II School for Information Technology, Computer Science Department, 11942, Jordan

### ARTICLE INFO

*Article history:*

*Received: 18 May, 2018*

*Accepted: 11 June, 2018*

*Online: 26 June, 2018*

*Keywords:*

*Distributed System*

*Dual core*

*Multicore*

*Quad core*

### ABSTRACT

*Multicore processor combines two or more independent cores onto one integrated circuit. Although it offers a good performance in terms of the execution time, there are still many metrics such as number of cores, power, memory and more that effect on multicore performance and reduce it. This paper gives an overview about the evolution of the multicore architecture with a comparison between single, Dual and Quad. Then we summarized some of the recent related works implemented using multicore architecture and show the factors that have an effect on the performance of multicore parallel architecture based on their results. Finally, we covered some of the distributed parallel system concepts and present a comparison between them and the multiprocessor system characteristics.*

### 1. Introduction

Multicore is one of the parallel computing architectures which consists of two or more individual units (cores) that read and write CPU instruction on a single chip [1]. Multicore processor can run multiple instructions at the same time which increase speed of program execution and performance.

Original processor has one core, dual core processor has two cores, quad core processor has four cores, hexa core has six, octa core has eight, deca contains ten cores and so on. Gordon Moore predicted many that number of cores on the chip will be doubled once in every 18 months keeping in mind performance and cost which is known as Moore's Law [2, 3, 4].

Each core in the multicore processor should not be necessarily faster than single core processor, but the overall performance of the multicore is better by handling many tasks in parallel [5].

The implementation of the multicore could be in different ways, homogeneous or heterogeneous depending on the application requirements. In the homogeneous architecture, all the cores are identical, and they break up the overall computations and run them in parallel to improve overall performance of the processor

[6]. The heterogenous cores have more than one type of cores, they are not identical, each core can handle different application. The later has better performance in terms of less power consumption as will be shown later [2].

In general, performance refers to the time needed to execute a given task, which in other words can be expressed as the frequency multiplied by the number of instructions executed per clock cycle as given in formula 1.

$$\text{Performance} = \text{Instructions executed Per Clock Cycle} * \text{Frequency} \quad (1)$$

Both frequency and instruction per clock cycle are important to processor performance which can be increased by the increasing of these two factors. Unfortunately, high frequency has some implications on the power consumption.

The power problem is very important, while performance still desired as predictions from the ITRS Roadmap [7] predicting a need for 300x more performance by 2022 with a chip with 100x more cores than current one [8] as shown in Figure 1.

$$\text{Power} = \text{Voltage}^2 * \text{frequency} \quad (2)$$

The multicore processor designer should balance between power (which related to the voltage and frequency) and instructions per

\*Ola Surakhi, University of Jordan, Email: ola.surakhi@gmail.com

cycle (IPC, which is related to the speed and throughput) when developing processors to get a high performance and power efficiency of the processor.

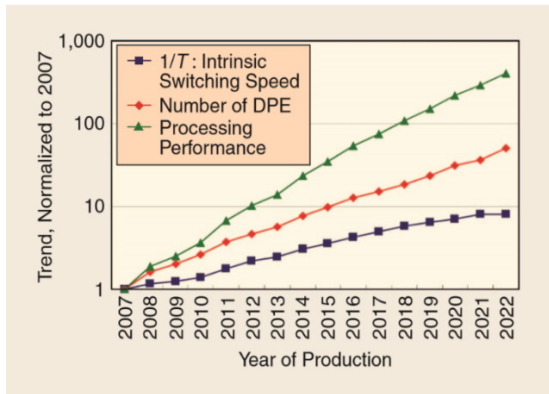


Figure 1: ITRS Roadmap for frequency, number of data processing elements (DPE) and overall performance [7, 8].

In this paper, an overview is given about multicore processor and its improvements, with focusing on introducing some of main constraints that effect on the performance of multicore. The rest of this paper is organized as follow: Section 2 gives a brief introduction about evolution of multicore architecture, Dual and Quad core with their advantages and disadvantages. Section 3 surveys some of the recent applications that had been implemented using multicore. Section 4 Describes the distributed systems, and section 5 gives the conclusion.

## 2. From single core to multi core

### 2.1. Single core processor

Single chip multiprocessors implement multiple processor cores on the same chip which provides a high performance with a shared memory. To understand the difference between performance of a single core and multicore processors, a brief detail is given for them below. Single CPU with single core refers to the processor that contains single core [9], it is simple, not used and not manufactured much. Figure 2 presents the main components of single core architecture.

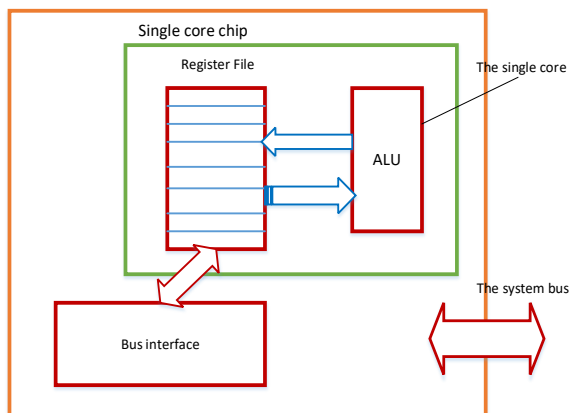


Figure 2: single CPU single core architecture [9]

A set of advantages and disadvantages for using single core are summarized below:

### Advantages of single CPU single Core architecture

- It takes less power to run single CPU with single core, for that it can be used for light processing such as OIT environment with no need for high powerful computer and at same time sensor node needs type of processor which does consume a lot of power and energy.
- Runs cooler: needs fewer number of fan for cooling.

### Disadvantages of using single CPU and single core

- Run slower: the advanced architectures that contain more than one core on a single processor is much faster than single CPU with single core processor.
- Freezing or can't run different modern applications such as application which have heavy computing, this type of application can't be executed using single CPU single core.
- The memory needs bigger cache size, latency still not in line with processors speed.

### 2.2. Multicore processor

Which refers to another advanced version of processor. It consists of more than one core of processor to increase the overall performance of processing. This version of processor called implies that contain two or more cores in the same physical package where each of these cores have its own pipeline and resources to execute different instructions without causing any problem. The main architecture for multicore processor is Multiple Instruction Multiple Data stream (MIMD). All of these threads can be executed at different cores on the same stream with same shared memory, for that these cores were implemented on the same computer instead of using single processor with single core shared with memory, as shown in Figure 3 [10].

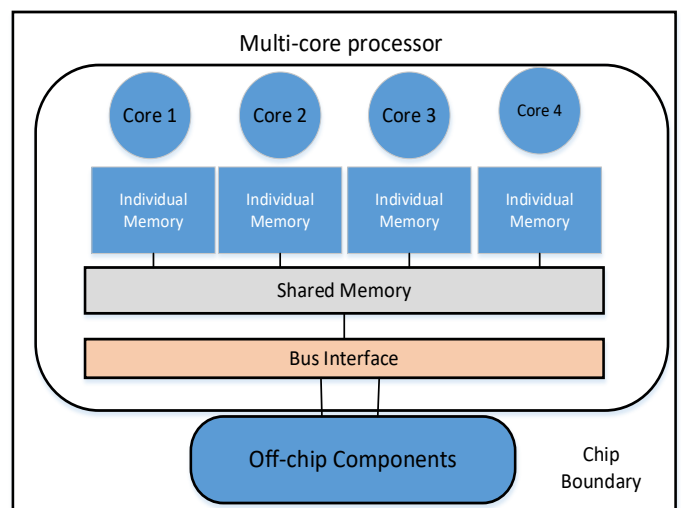


Figure 3: Multicore processor with shared memory [10]

Based on the idea of multicore processor which contains several processors ‘cores’ and packages inside single physical processor it increases overall performance of machine by allow more than one instruction to be executed at the same unit of time. As a result, simple multicore processors have better performance than complex single core processors [11].

### Main advantages of multicore processor

- Allows cache coherency circuitry to operate at higher clock rate than is possible.
- Combining equivalent CPU on a single die can increase and improve the performance of cache snoop significantly.
- Many applications such as Customer Relationship Management, larger databases, Enterprise Resource Planning, e-commerce and virtualization can get a big benefit of multicore.

### Multicore Challenges

There are a set of problems that arise when the multiple cores are implemented on a single chip [12]. Some of them are described below.

- Power and temperature problem, having multicores on the same chip will generates more heat and consumes more power than single CPU on a chip which can be solved by turning off unused cores to limit the amount of power.
- The need for enhancing in the interconnection network and bus network: managing the time required to handle memory requests to enhance speed and reducing communications between cores.
- The need of programming in parallel environment: the programmer needs to learn how to write a code for running a program on multicore instead of single core. As the total amount of cores on single chip is increasing as per Moore’s law, the quantity of parallelism should be doubled during development of the software.
- Denial of memory services, cores on the same chip share same the DRAM memory system and when multiple programs or branch of program run at multicore at the same time it will interfered with other memory access request.
- Multithread problem for multi core technology, starvation problem may occur if program developed to be run on a single thread while environment is multithread. This will cause a starvation where one thread is working, and others are idle.
- cache coherence problem, where different cores write and read on cache memory

There are different versions of multicore processor start from dual core technology [11], the CPU contains dual core, that is having two actual execution units on a single integrated circuit [9]. This refers to initial version of improvement of multicore technology

which has a lot of advantages and strongest point over old version of single core technology as it is easy for switching from single core processor to dual core processor with greater performance and low electrical costs comparing with single core technology. For everyone who can follow information technology industry knows that it is difficult and costs to increase the speed scale of clock rate for any processor, which means higher clock frequency and needs more heat. This will increase temperature of the chip which in turn needs to be cooling to reduce it [12]. In the dual core processor, the cores can execute programs in parallel which increase performance and at the same time the cores are not clocked at higher frequency which consumes more energy efficient. The design of the cores allows those who do not have any job (idle) to power down and when needed it can be powered up again which save power at all. The energy efficient performance for the dual core is shown in Figure 4, which shows how reduced clock frequency by five has more performance comparing with single core which consumes the same power but with maximum frequency.

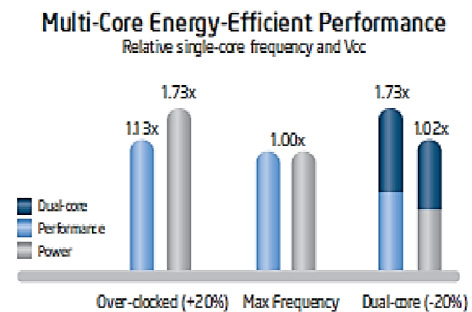


Figure 4: Dual core processor at 20% reduced clock frequency effectively delivers 73% more performance while approximately using the same power as a single core processor at maximum frequency. [13,14]

Another advantage of using dual core processor rather than using single core processor refers to reducing the overall cost of processing, before dual core investment the users need to build dual processor units that had computing *power same as single dual core processor* with double cost of single computer with dual core.

Figure 5 presents the architecture of dual core processor that can execute different thread. Cores run without interference with each other.

As we have advantages for dual core technology there are disadvantages as follows

Some programs are not designed to run on more than one core, a single core processor will run such program faster than dual core. Running that programs on dual core will decrease performance.

If a single core CPU has a significantly greater clock speed than dual core CPU, then single core CPU can outperform dual core CPU.

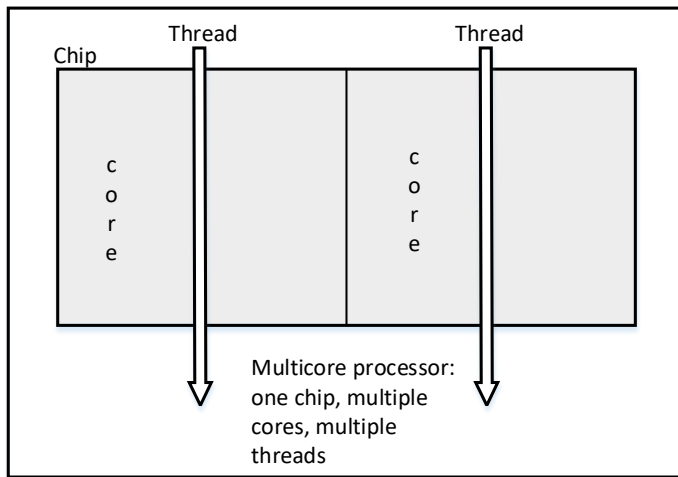


Figure 5: Dual core processor

### 2.3. Quad core technology

This refers to another version of multicore technology which contains four cores on the same chip or single processor which allows more threads to be executed in parallel which will lead to have higher performance than single and dual core processor. Figure 6 shows the architecture for quad core processor which allows four threads to be executed at the same time.

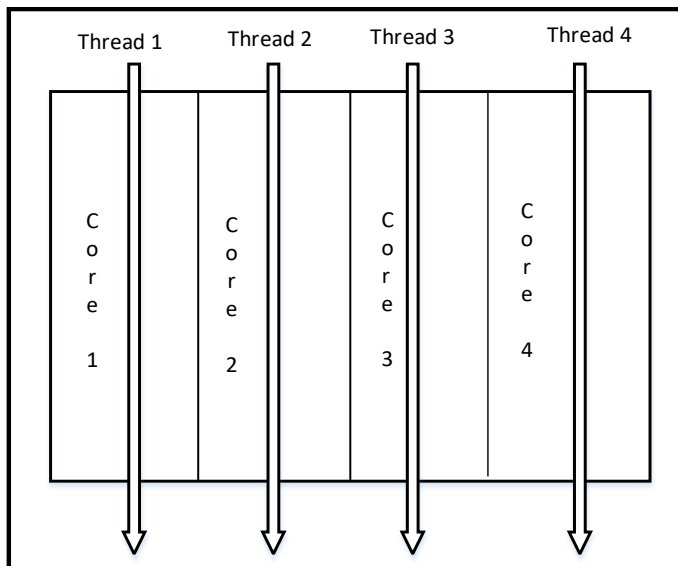


Figure 6: Quad Core Architecture which contains four cores.

### Main advantages of using quad core technology

First advantage refers to increase performance and power of execution instructions which will lead to execute up to four processes at the same time.

Another advantage of quad core refers to the ability to run different programs which need heavy processing such as anti-virus programs and graphical applications which need high processing for that these technologies can be run smoothly

Other advantages over all previous multicore technology refer to less heat and power consuming because this technology was designed small and efficient

Other advantage of using quad core technology refers to Use for long term: The problem with Moore's Law is that it practically guaranteed that your computer would be obsolete in about 24 months. Since few software programs are programmed to run on dual core, much less quad core, these processors are actually way ahead of software development.

### Disadvantages of using Quad core technology

The main disadvantage of quad core technology refers to high power consuming which is higher than dual core technology

A challenge in using quad core technology refers to the ability of the software to investigate parallel quad core technology. The software cannot take the benefit of parallelism in quad core technology if the program has a large sequential part that cannot be separated.

In order to compare between single core and multiprocessor parallel architectures performance, a set of metrics should be considered. Table 1 introduces such comparison.

### 3. Applications and algorithms build using multicore architecture

In order to show the efficiency of using multicore parallel architecture over sequential one, this section will give a survey of the state-of-the-art of parallel implementations proposed for a particular application using multicore architecture.

In [15], authors proposed a parallel implementation for Proclus algorithm using multicore architecture. The results of the implementations showed an enhancement on the performance of the algorithm in terms of running time for a large dataset.

In [16], a parallel implementation for maxflow problem using multicore had been proposed by dividing graph in to a set of augmenting paths where each path can run on a single thread. The results showed a good enhancement according to the execution time comparing with sequential version.

A parallel implementation for RSA algorithm using multicore architecture had been proposed in [17]. The main contribution of the work is the enhancing in speed of the algorithm when using a hybrid system with multicore CPU and GPUs (graphics processing unit that is composed of hundreds of cores) with variable key size.

Authors in [18] implemented merge sort which is a divide-and-conquer sorting algorithm on Java threading application programming interface (API) environment which allows programmer to directly implements threads in the program. The results of the experiment were tested against different data size

with different number of processors, and showed a good performance compared with sequential version.

In [19], authors implemented a parallel dynamic programming algorithm on multicore architecture in cognitive radio ad hoc networks (CRAHNS), the results showed a significant gain in terms of execution time.

In [20], authors implemented a parallel version of Breadth First Search (BFS) algorithm on multicore CPUs which show a great enhancing in the performance over current implementation as the size of the graph increases.

In [21], authors developed a parallel algorithm for the Euclidean Distance Map (EDM) which is a 2-D array and each element in it stores a value with its Euclidean distance between it and the nearest black pixel. The implementation was done on two parallel platforms multicore processors and GPU. The results showed an enhancing in the speed when using multicore platform for a specific input size 10000x10000 image size over GPU and sequential version

Authors in [22] implemented a sequential and parallel version of Viola-Joins face detection on multicore platform, the results that increasing number of cores for face detection in the image will increase speed up even though for big size image.

In [23], authors analyses the performance improvement of a parallel algorithm on multicore systems and show that a significant speed up achieved on multicore systems with the parallel algorithm.

In [24], authors mapped 4 implementations for Advanced Encryption Standard (AES) cipher algorithm on a fine-grained many-core system with eight cores online and six offline key expansions for a small design and 107 and 137 cores online and offline respectively for a largest design. The proposed design showed a higher throughput per unit of chip area and 8.2-18.1 times higher energy efficiency.

A summary for the results concluded from above mentioned works is shown in table2.

### 3.1. Discussion and Analysis

Many applications had been implemented in parallel using multicore platform to solve large and small problems as shown before. Some works implemented large algorithms in parallel such as Genetic [25], PSO [26], Tabu search algorithm [27] and more.

In order to evaluate the performance of such parallel implementations, there are some aspects that affect on the performance of the algorithms and need to be taken into consideration when evaluating parallel algorithm results. The main reason of using multicore architecture is to increase speed of the running program by dividing it into small pieces and run each piece independently on a single core. This means as many cores we have, will get faster execution! But it is not the proper

scenario that happens at real, adding more processors does not implies faster execution, as there are many other aspects that effect on the execution speed of the processor, like problem size, data input, communications between cores which is related to the interconnection network and computations needed on each core which is related to the problem. On the other hand, more cores on the processor need more power consumption which will reduce performance of the system which make companies like IBM, Intel to develop an enhanced version of the multicore chip that reduce power consumption.

The cache utilization in the multicore is shared between cores which increase cache access time, and this involve a higher percentage to access memory which in turn effects on the execution and performance. It can be reduced by implementing cache levels in a hierarchical design and including more cores in the processor which will decrease access time to the cache.

Degree of parallelism on each core allows instructions to run simultaneously to increase performance by increasing throughput. Pipelining is one of the techniques used to execute instructions in stages where number of instructions in the pipeline equal to number of stages. On the thread level, achieving parallelism is different here where program is divided into small pieces (threads) that run simultaneously in a synchronized manner such as Round Robin. The degree of parallelism could be decreased here when a starvation event happens where one thread take CPU time and keep all other waiting for execution which decrease performance.

Another problem that may happen here is Race condition when two threads need to access the same resource which also cause the system to get in to a deadlock.

Power, high power consumption causes the chip to have much heating and may be unreliable to do calculations and need much cooling. Many approaches had been used to reduce overheating in cores such as Stop & Go, Thread Migration(TM), and Dynamic Voltage and Frequency Scaling (DVFS) [28]. The main source that cause heating in the core is frequency which is not easy to optimize as it proportional to both performance and power.

Problem size, small input size leads to slowdown parallelism, because the communication will be high comparing to the computation done by each core. On the other hand, large input size may effect on the speed up and efficiency. As the problem size increases, the speed up and efficiency will increase to a specific limit depending on number of cores and cache size.

Many other aspects may effect on the performance of the multicore architecture, such as efficiency (keeping all cores busy during execution), architecture design issues, RAM size, parallel programming language, etc.

## 4. Distributed memory computers

It is a message-passing system where each processor has its own memory and all processors are connected by each other through a high-speed communication link. As each processor has its own

memory, the memory addresses in one processor do not map to another processor. The data exchange between processors explicitly by message-passing. The architecture of the distributed memory is shown in Figure 7.

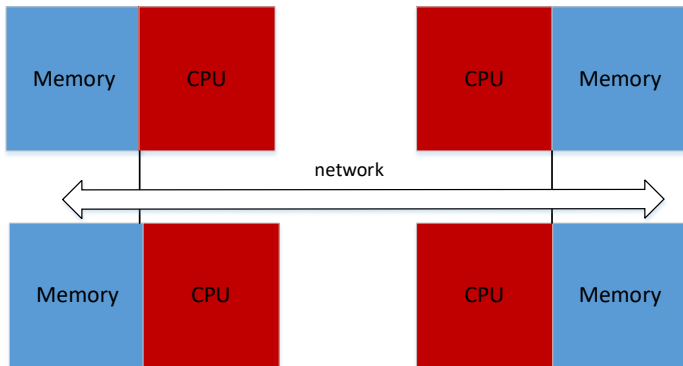


Figure 7: Distributed memory architecture [29]

In a distributed system there is no shared common physical clock which inherits asynchrony amongst the processors that do not share the same physical clock. And no shared memory as each computer in the distributed system has its own memory, the nodes can communicate between each other through message passing [29].

#### Advantages of distributed memory architecture

- Memory is scalable with the number of processors.
- Increase the number of processors, the size of memory increases proportionally.
- Each processor can rapidly access its own memory without interference and without the overhead incurred with trying to maintain cache coherence.
- Cost effectiveness: can use commodity, off-the-shelf processors and networking.

#### Disadvantages of distributed memory architecture

- Difficult to program: Programmer has to handle data communication between processors.
- Non-uniform memory access (NUMA) times.

It may be difficult to map existing data structures, based on global memory, to distributed memory organization.

Examples for parallel distributed computers are:

1. Cluster Computing: a set of parallel and distributed computers that are connected through a high-speed network. They work together in the execution of compute intensive and data intensive tasks that would be not feasible to execute on a single computer [30]. The cluster has a high availability by keeping on redundant nodes to be used when some of the existing failed which increase performance of the system [31]. Each cluster consists of multiple computers that are connected with each other and share computational work load as a single virtual computer. The work is distributed among

different computers which results in balancing load between them.

2. Grid computing: is a type of parallel and distributed computers which combines a set of computers from different domains to solve a problem and reach a common goal [32]. Grid computing divides the program between different computers by using a middleware to do that.
3. Cloud computing: is type of parallel and distributed system that consists of a set of interconnected and virtualized computers that provide a service to be delivered as an application i.e. infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) etc [33].

#### 4.1. Parallel Distributed Algorithm Models

The parallel algorithm model is a technique used to decompose tasks properly, mapping them by applying an appropriate strategy to decrease overhead [34, 35, 36]. A set of parallel distributed algorithms is listed in this section.

1. The Data-Parallel model: this model is simple which assigns tasks to the computing elements statically where each task does the same operations but with different data [36]. The parallelism occurs as the processors operate similar operations, but the operations may be executed in phases having different data [36, 37].
2. The Task Graph Model: this model used task dependency graph to map tasks to the processors [36, 37].
3. The Work Pool Model: is a dynamic mapping technique that assigns tasks by centralized or decentralized fashion to achieve load balancing. whenever the process becomes available, it is added to the global work pool [36, 37].
4. The Master-Slave Model: in this model there is one node designated as a master while all other nodes are slaves [38, 39]. The master node generates work and distributes it among all other nodes. The slaves do the work and master node collects then results.
5. The Pipeline or Producer-Consumer Model: in this model, the data is passed through a pipeline which consists of a set of stages where the output from one stage becomes an input to the next one to achieve overlapping [40].
6. Hybrid model: which combines two models to form a new one.

Comparison between distributed system and multiprocessors is shown in table 3.

## 5. Conclusion

The number of cores in the multiprocessor chip is increasing very rapidly every new generation of CPU. With each new generation the performance of it is increased. Many factors affect on the performance of multicore processor. This paper gives an overview for the evolution of multicore, compares between Dual and Quad, and survey a number of the recent works that had been implemented using multicore platform. The results of presented works show that increasing number of cores in the multicore

processors does not mean to get higher performance nor a speedup. The speed up of the system will be increased to a number of cores, where after that value the performance of the system will not be increased more.

Other factors have an influence on the efficiency and performance of multicore system such as RAM size, problem size, degree of parallelism, power and more as mentioned before.

A comparison between multiprocessor and distributed parallel system is made, it shows that distributed system is scalable, has a high degree of transparency but harder to be programmed, while multiprocessor system is not scalable, has a high degree of transparency and easy to be programmed.

As more applications and programs are developed to take advantage of multicore architecture, the designing and performance issues will be a challenge for most of the commercial industrial companies.

## References

- [1] Lizhe Wang, Jie Tao, Gregor von Laszewski, Holger Marten. 2010, "Multicores in Cloud Computing: Research Challenges for Applications", *Journal of Computers*, Vol 5, No 6 (2010)
- [2] Parkhurst, J., Darringer, J. & Grundmann, B. 2006, "From Single Core to Multi-Core: Preparing for a new exponential", *Computer-Aided Design*, 2006. ICCAD '06. IEEE/ACM International Conference on, pp. 67.
- [3] Schaller, R.R. 1997, "Moore's law: past, present and future", *Spectrum*, IEEE, vol. 34, no. 6, pp. 52-59.
- [4] Sodan, A., Machina, J., Deshmeh, A., Macnaughton, K. & Esbaugh, B. 2010, "Parallelism via Multithreaded and Multicore CPUs", *Computer*, vol. PP, no. 99, pp. 1-1.
- [5] D. Geer, "Chip Makers Turn to Multicore Processors," *Computer*, vol. 38, pp. 11-13, 05, 2005
- [6] Blake, G., Dreslinski, R.G. & Mudge, T. 2009, "A survey of multicore processors", *Signal Processing Magazine*, IEEE, vol. 26, no. 6, pp. 26-37.
- [7] International Technology Roadmap for Semiconductors, "International technology roadmap for semiconductors—System drivers," 2007 [Online]. Available: [http://www.itrs.net/Links/2007ITRS/2007\\_Chapters/2007\\_SystemDrivers.pdf](http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_SystemDrivers.pdf)
- [8] Geoffrey Blake, Ronald G. Dreslinski, and Trevor Mudge, "A Survey of Multicore Processors", *IEEE SIGNAL PROCESSING MAGAZINE* NOVEMBER 2009
- [9] A. Vajda, *Programming Many-Core Chips, Multi-core and Many-core Processor Architectures*, DOI 10.1007/978-1-4419-9739-5\_2, © Springer Science+Business Media, LLC 2011.
- [10] IRakhee Chhibber, IIDr. R.B.Garg, *Multicore Processor, Parallelism and Their Performance Analysis*, *International Journal of Advanced Research in Computer Science & Technology (IJARCST)* 2014.
- [11] B. RAMAKRISHNA RAU AND JOSEPH A. FISHER, "Instruction-Level Parallel Processing" History, Overview, and Perspective, *Journal of Supercomputing*, 7, 9-50 (1993) Kluwer Academic Publishers, Boston. Manufactured in The Netherlands
- [12] Mark T. Chapman, "The Benefits of Dual-Core Processors in High-Performance Computing", *Supercharging your high-performance computing* June 2005
- [13] R. Ramanathan. Intel multi-core processors: Making the move to quad-core and beyond. *Technology@Intel Magazine*, Dec 2006.
- [14] Balaji Venu, "Multi-core processors - An overview", <https://arxiv.org/ftp/arxiv/papers/1110/1110.3535.pdf>
- [15] M.Khanafseh, O.Surakhi & Y.Jaffal, "Parallel Implementation of "Fast algorithm for Project Clustering" on Multi Core Processor using Multi Threads", *Journal of Computer Science IJCSIS* June 2017 Part I.pdf, (pp. 248-255)
- [16] Mohammed Y. Alkhanafseh, Mohammad Qataweh, Hussein A. al Ofeishat, "A Parallel Chemical Reaction Optimization Algorithm for MaxFlow Problem (pp. 19-32), *Journal of Computer Science IJCSIS* June 2017 Part I.pdf, Vol. 15 No. 6 JUN 2017
- [17] Heba Mohammed Fadhil, Mohammed Issam Younis, "Parallelizing RSA Algorithm on Multicore CPU and GPU", *International Journal of Computer Applications* (0975 – 8887) Volume 87 – No.6, February 2014
- [18] Soha S. Zaghoul, PhD1, Laila M. AlShehri2, Maram F. AlJouie3, Nojood E. AlEissa4, Nourah A. AlMogheerah5, "Analytical and Experimental Performance Evaluation of Parallel Merge sort on Multicore System", *International Journal Of Engineering And Computer Science* ISSN:2319-7242 Volume 6 Issue 6 June 2017, Page No. 21764-21773
- [19] Badr Benmammam, Youcef Benmouna, Asma Amraoui, Francine Krief, "A Parallel implementation on a Multi-Core Architecture of a Dynamic Programming Algorithm applied in Cognitive Radio Ad hoc Networks", *International Journal of Communication Networks and Information Security (IJCNIS)*, Vol. 9, No. 2, August 2017
- [20] Sungpack Hong , Tayo Oguntebi , Kunle Olukotun , "Efficient Parallel Graph Exploration on Multi-Core CPU and GPU" ,2011 International Conference on Parallel Architectures and Compilation Techniques.
- [21] D.Man, K.Uda, H.Ueyama, Y.Ito, K.Nakano, "Implementation of Parallel Computation of Euclidean Distance Map in Multicore Processors and GPUs", 2010 First International Conference on Networking and Computing
- [22] Subhi A. Bahudaila and Adel Sallam M. Haider, "Performance Estimation of Parallel Face Detection Algorithm on Multi-Core Platforms", *Egyptian Computer Science Journal* Vol. 40 No.2 May 2016, ISSN-1110-2586
- [23] Sheela Kathavate, N.K. Srinath, "Efficiency of Parallel Algorithms on Multi Core Systems Using OpenMP", *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 3, Issue 10, October 2014
- [24] M.Sambasiva Reddy, P.James Vijay, B.Murali Krishna, "Design and Implementation of Parallel AES Encryption Engines for Multi-Core Processor Arrays", *International Journal of Engineering Development and Research*, 2014 IJEDR | Volume 2, Issue 4 | ISSN: 2321-9939
- [25] A.J.Umbarkar,M.S.Joshi,Wei-ChiangHong, "Multithreaded Parallel Dual Population Genetic Algorithm (MPDPGA) for unconstrained function optimizations on multi-core system", *Applied Mathematics and Computation*, **Volume 243**, 15 September 2014, Pages 936-949
- [26] Abdelhak Bousbaci , Nadjet Kamel , "A parallel sampling-PSO-multi-core-K-means algorithm using mapreduce",*Hybrid Intelligent Systems (HIS)*, 2014 14th International Conference on 14-16 Dec. 2014
- [27] Ragnar Magnús Ragnarssona, Hlynur Stefánssona, Eyjólfur Ingi Ásgeirssona, "Meta-Heuristics in Multi-Core Environments", *Systems Engineering Procedia* 1 (2011) 457–464
- [28] Ransford Hyman Jr, "Performance Issues on Multi-Core Processors".
- [29] Jelica Protić, Milo Tomagević, Veljko Milutinović, "A Survey of Distributed Shared Memory Systems", *Proceedings of the 28th Annual Hawaii International Conference on System Sciences* - 1995
- [30] Brijender Kahanwal, Tejinder Pal Singh, "The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle", *International Journal of Latest Research in Science and Technology* ISSN (Online):2278-5299 Vol.1, Issue 2 :Page No.183-187 ,July-August(2012)
- [31] "Cluster Computing", [http://en.wikipedia.org/wiki/Cluster\\_computing](http://en.wikipedia.org/wiki/Cluster_computing).
- [32] K. Krauter, R. Buyya, and M. Maheswaran, "A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing", *Jr. of Software Practice and Experience*, 32,(2), pp. 135-164, 2002.
- [33] Seyyed Mohsen Hashemi, Amid Khatibi Bardsiri (MAY 2012) "Cloud Computing Vs. Grid Computing", *ARPN Journal of Systems and Software*, VOL. 2, NO.5
- [34] B Barney, *Introduction to Parallel Computing*, Retrieved from Lawrence Livermore National Laboratory: [http://computing.llnl.gov/tutorials/parallel\\_comp/2010](http://computing.llnl.gov/tutorials/parallel_comp/2010)
- [35] I Ahmad, A Gafoor and G C Fox, *Hierarchical Scheduling of Dynamic Parallel Computations on Hypercube Multicomputers*, *Journal of Parallel and Distributed Computing*, 20, 19943, 17-329.
- [36] A Grama, A Gupta, G Karypis and V Kumar, *Introduction to Parallel Computing*, Publisher: Addison Wesley, uJanuary 2003
- [37] K Cristoph and K Jorg, *Modles for Paralel Computing: Review and Perspectives*, *PARSMiteilungen* 24, Dec. 2007, 13-29
- [38] A Clamatis and A Corana, *Performance Analysis of Task based Algorithms on Heterogeneous systems with message passing*, In *Proceedings Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 5th European PVM/MPI User's Group Meeting, Sept 1998.
- [39] D Gelernter, M R Jourdenais and Kaminsky, *Piranha Scheduling: Strategies and Their Implementation*, *International Journal of Parallel Programming*, Feb 1995, 23(1): 5-33
- [40] S H Bokhari, *Partitioning Problems in Parallel, Pipelined, and Distributed Computing*, *IEEE Transactions on Computers*, January 1988, 37:8-57

Table 1: A comparison between Single core, Dual core and Quad core

<b>Difference criteria</b>	<b>Single core</b>	<b>Dual Core</b>	<b>Quad Core</b>
<b>Description</b>	Use Single Core processor to process different operations, this type of processor used by many devices such as smartphones	Processor with two cores on a single integrated circuit, which acts as a single unit, each of these cores have a separated controller and cache which allow them to work faster.	Contains two dual core processor on a single chip, with all its property and performance.
<b>Execution speed</b>	Runs slower based on sequential execution for tasks.	Perform tasks faster based on parallel execution for impendent tasks	One of the best systems of multi-tasking, based on that it can execute a lot of tasks while other systems start up.
<b>Multi task possibility</b>	Possible by pre-emptive round robin scheduling	Possible	Possible
<b>Power usage</b>	Less Power Usage, to execute single core process processor take less power if comparing with others.	Wastes power more than single core because it need more cooling	Low battery life and can't be used for battery powered devices.
<b>Application support</b>	Word processing, surfing, browsing checking email, mobile applications, sensor applications and IOT devices	Flash-enabled web browsing.	Voice GPS systems and multi-player and video editing.
<b>Parallel execution for tasks</b>	Can execute other process while still work on first task but this depends on the application	It has two complete two execution units which can execute different instructions completely if there is no dependency	This architecture consists of two separated dual core which can execute double of what dual core can execute.
<b>Need for cooling</b>	Using less power which produces less heat which means that no need for advance cooling	Need for cooling but not much cooling	Need for advance cooling which consume a lot of energy



Table 2: summary for different parallel implementations results

Research Ref. No	Problem area	Conclusion
[15]	Clustering algorithm, "Proclus	Using parallel implementation, gained a large enhancement in terms of speed up.
[16]	Solving maxflow problem using CRO algorithm	The level of speed up enhancement is large from sequential level to parallel with two cores, then the enhancement is not increased much as the input size increases.
[17]	Security, RSA algorithm	Many core architectures are more powerful for heavy computations with high speed up. Throughput can affect on the speed up.
[18]	Merge Sort	Increasing number of the threads leads to an increase in the problem size that can be handled which reflects scalability of parallel algorithm
[19]	dynamic programming algorithm applied in cognitive radio ad hoc networks	Using more threads does not necessarily decrease execution time. Indeed, with a large number of threads, we have more parallelism and therefore more communications that impact the execution time.
[20]	Graph, BFS algorithm	single high-end GPU performs as well as a quad-socket high-end CPU system for BFS execution; the governing factor for performance was primarily random memory access bandwidth
[21]	Compute Euclidean Distance Map (EDM)	Considering many programming issues of the GPU system such as coalescing access of global memory, shared memory bank conflicts and partition camping can affect on acceleration.
[22]	Image processing, Face Detection algorithm	Increasing number of cores for face detection will increase speed up for face detection for big size of images
[23]	Matrix multiplication algorithm	Execution time can be reduced when using parallel implementation but after a certain input size
[24]	Security, AES algorithm	The fine-grained many core system is recommended for AES implementation in terms of energy efficiency.

Table 3: Comparison between distributed system and multiprocessors

	Multiprocessor Parallel System	Distributed Parallel System
Scalability	Harder	Easier
Programmability	Easier	Harder
Degree of transparency	High	Very high
Architecture	Homogeneous	Heterogeneous
Basic of communication	Shared memory	Message passing