SWOT Analysis of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI and their Usage in Various Companies

Shajil Kumar P. A. [1] & Srinivasa Rao Kunte [2]

[1] Research Scholar, Institute of Computer Science and Information Science,
Srinivas University, Mangaluru, Karnataka, India,
Assistant Professor, Vidyalankar School of Information Technology, Mumbai, Maharashtra, India, OrcidID: 0000-0003-1250-4882; E-mail: shajilkumar.pa@gmail.com

[2] Research Professor, Srinivas University, Mangaluru, Karnataka, India,
OrcidID: 0000-0002-5062-1505; Email: kuntesrk@gmail.com

Subject Area: Computer Science.
Type of the Paper: Exploratory Research.
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed In: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.10451939
Google Scholar Citation: IJAEML

How to Cite this Paper:
Shajil Kumar, P. A. & Kunte, S. R. (2023). SWOT Analysis of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI and their Usage in Various Companies. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *7*(4), 300-319. DOI: https://doi.org/10.5281/zenodo.10451939

International Journal of Applied Engineering and Management Letters (IJAEML)
A Refereed International Journal of Srinivas University, India.

Crossref DOI: https://doi.org/10.47992/IJAEML.2581.7000.0206

Received on: 09/08/2023
Published on: 30/12/2023

Disclaimer: The scholarly papers as reviewed and published by Srinivas Publications (S.P.), India are the views and opinions of their respective authors and are not the views or opinions of the S.P. The S.P. disclaims of any harm or loss caused due to the published content to any party.

SWOT Analysis of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI and their Usage in Various Companies

Shajil Kumar P. A. [1] & Srinivasa Rao Kunte [2]

[1] Research Scholar, Institute of Computer Science and Information Science,
Srinivas University, Mangaluru, Karnataka, India,
Assistant Professor, Vidyalankar School of Information Technology, Mumbai, Maharashtra, India, OrcidID: 0000-0003-1250-4882; E-mail: shajilkumar.pa@gmail.com

[2] Research Professor, Srinivas University, Mangaluru, Karnataka, India,
OrcidID: 0000-0002-5062-1505; Email: kuntesrk@gmail.com

ABSTRACT

Purpose: *Parallel Processing is the technique of achieving High Performance Computing (HPC) with parallel execution of programs that are synchronized during the execution time. This research paper studied the companies which use Parallel Processing techniques in their projects and products along with the identification of major Application Program Interfaces (APIs) that are used to achieve parallelism. The major aim of this research work is to perform the SWOT analysis of Parallel Processing APIs to identify the importance of each one from the company perspective.*

Design/Methodology/Approach: *The research method adopted to accomplish the SWOT Analysis of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI and their Usage in Various Companies is qualitative and exploratory in nature. Systematic literature review of different companies that use Parallel Processing techniques to build and develop parallel programs is done during this research work.*

Findings/Results: *Parallel Processing constructs can be used to solve various problems in the six major application domains as: - Computational Finance & Business Economics, Artificial Intelligence, Machine Learning (ML), Data Science, Numerical Analysis and Design of Algorithms. Major Parallel Processing APIs used in companies are: - CUDA, OpenCL, OpenMP and MPI for implementing the problems with parallel execution. Foremost companies that use Parallel Processing APIs are studied and various applications, systems, models, and projects that are developed using Parallel Processing techniques are listed. SWOT Analysis is performed on all four Parallel Processing APIs and its SWOT(Strengths-Weaknesses-Opportunities-Threats) are identified.*

Originality/Value: *Listing of SWOT Analysis (Strengths-Weaknesses-Opportunities-Threats) of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI.*

Paper Type: *Company Analysis research paper*

Keywords: Parallel Processing, SWOT Analysis, Company Analysis, CUDA, OpenCL, OpenMP, MPI

1. INTRODUCTION :

Parallel Processing is the concept of using multiple processors or computing capabilities to execute multiple tasks simultaneously to achieve faster results. Overall computation capability of a particular process can be increased by distributing CPU-intensive processes amongst multiple processing units; hence increasing the efficiency of the program. Parallel Processing methods can be achieved either using specialized hardware or software algorithms that allow multiple tasks to be executed simultaneously on a single processor. Parallel Processing works by breaking down complex computations into smaller manageable tasks; and then processes these tasks simultaneously to increase the productivity of the overall system and computation capabilities. This research paper is a company analysis-based research paper that aims at first in identifying the major application areas of Parallel

Processing. It also aims in identifying the most used APIs by companies in their Parallel Processing applications and projects. Also study the background company details of the identified Parallel Processing APIs. The paper also aims at performing SWOT Analysis (Strengths, Weaknesses, Opportunities, and Threats) of these APIs.

2. OBJECTIVES :

This research paper has the following objectives:

(1) To identify major application domains of Parallel Processing.
(2) To study the company details of four major Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI
(3) To study different companies that use Parallel Processing APIs and list their major projects that use Parallel Computing techniques.
(4) To perform SWOT(Strengths-Weaknesses-Opportunities-Threats) Analysis of four major Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI.

3. METHODOLOGY :

This research paper is prepared with the qualitative exploratory research methodology.

- To identify major companies that use Parallel Processing APIs and major projects that involve Parallel Computing techniques; various company websites are referred to know how well these methods are used in the company's perspectives.
- The major application domain of Parallel Processing is identified through qualitative exploratory research methodology and literature review of existing research papers.
- Company particulars of four major Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI are done with the support of the corresponding APIs websites.
- Major objective of the paper is to perform SWOT Analysis of four Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI, which is done with extensive literature review of various research papers that have already been published in the domain of Parallel Processing.
- As Parallel Processing APIs are vital in managing scalable data systems, SWOT identification of these APIs will give new insights in this emerging domain of knowledge.

Systematic literature review is done by considering the purpose of study using various online resources like ResearchGate, IEEE Xplore and Google Scholar.

4. LITERATURE REVIEW :

Detailed literature review is done on various Parallel Processing techniques adopted by companies in developing applications, systems, models, and projects that can enhance the computing power in association with the sequential program execution and listed in Table 1.

Systematic literature review is conducted on various products & projects that are designed in parallel and are listed in major areas like: -

- Computational Finance & Business Economics – Parallel processing can enhance the performance of various fields of Computational Finance like: - Derivative Pricing, Portfolio Optimization and Risk Management. In the domain of Business Economics, parallel applications can enhance Market Research, Supply Chain Management, Econometric Modeling and Optimization.
- Artificial Intelligence - Parallel processing techniques can be used in Natural Language Processing, Hyper parameter Tuning, Convolutional Neural Networks and Deep Learning Training
- Machine Learning (ML) - Video Haze Removal, Linear Algebra Software and K- means Clustering are different machine learning methods that can be done in parallel.
- Data Science - Option Pricing using Multinomial Tree Approach and Model to prevent train accident prevention are some of the scenarios where parallel applications work in the domain of data science.
- Numerical Analysis – Techniques like: - Finite Volume Method, Alternating Direction Implicit, Phase Field Model, Parallel Algorithms for Back Substitution, Conjugate Gradient and Gauss Seidel and Discrete Orthogonal Moments can be done in parallel to enhance the execution time.

- Design of Algorithm – Parallel algorithms can increase the speed and efficiency of the algorithmic procedures like: - Divide and Conquer, MapReduce Paradigm, Parallel Search Algorithms, Graph Algorithms, Parallel Sorting, Parallel Optimization Algorithms.

Table 1: Scholarly literature review on Parallel Processing applications, systems, models and projects

S. No.	Area & Focus of the Research	Outcome of the Research	Reference
1	To develop parallel algorithms for autonomous vehicles.	A parallel algorithm to analyze spatial data is developed and its execution time is compared with a sequential method.	Oh, B. W. (2019). [1]
2	Development of parallel method for Finite Volume Method.	Parallel execution using GPU provides better results in comparison with OpenMP.	Afzal, et al., (2020). [2]
3	To parallelize Computer Vision procedures.	Efficient Computer Vision algorithm using CUDA is developed, and its execution time is verified with different image sizes.	Afif, et al., (2020). [3]
4	Parallel implementation of Alternating Direction Implicit using cyclic reduction method.	Uses of CUDA and OpenCL in solving the problem related to oil recovery are listed.	Imankulov, et al., (2021). [4]
5	Parallel method implementation for video haze removal.	The result shows that parallel implementation is 226 times faster in comparison with CPU processing for the problem of video haze removal.	Wu, et al., (2020). [5]
6	To implement 3D graphics features using OpenCL, without dedicated hardware devices for the graphics.	Designed OpenCL implementation for 3D graphics features and verified its efficiency.	Kim, et al., (2021). [6]
7	To analyze the features of OpenCL using C++ programming model.	Cost effective massive Parallel Processing methods for big data systems are analyzed.	Shin, et al., (2020). [7]
8	To use OpenCL parallel technology for the Phase Field Model.	Scale of computing power of parallel techniques is verified to identify the problem of a single processor.	Ma, C., et al., (2022). [8]
9	Implementation of multi asset option pricing using multinomial tree approach.	Developed option pricing model using the concept of programmable gate array method that executes more than 40 times faster than a single processor system.	Mahony, et al., (2022). [9]
10	To develop Open IoT that provides various IoT services using Tacit Computing technology.	Evaluated the execution time for operations like Darknet and Fourier Transform.	Yamato, Y. (2020). [10]
11	To develop a parallel image encryption algorithm named HCMO to enhance security, encryption efficiency and processing speeds.	Experimental results depict HCMO provides better security in comparison to general purpose algorithms.	You, et al., (2020). [11]

12	To develop a parallel algorithm for distributed generation (DG) allocation problems that are multiple objectives in nature	Parallel implementation helps in using Monte-Carlo simulation for solving distributed generation (DG) allocation problems.	Abdelaziz, et al., (2019). [12]
13	To develop parallel Finite volume method (FVM) code using OpenMP.	Performance of different OpenMP parallel methods is studied on different operating environments.	Afzal, et al., (2020). [13]
14	To implement parallel application library for Linear Algebra.	Verification of Execution Analysis is performed on IBM and Intel machines.	Dongarra, et al., (2019). [14]
15	Implementation of Image Block Representation (IBR) for binary images using OpenMP.	High level performance is measured for IBR implementation in OpenMP, on a multicore computer.	Spiliotis, et al., (2020). [15]
16	A new method of Parallel sorting method named (MPDMSort) is developed and its execution on shared memory systems is done.	Reseach results show that MPDMSort is faster than general merge sort with scalable data.	Ketchaya, et al., (2023). [16]
17	To improve the 3D - Discontinuous Deformation analysis with the help of parallel technique.	Developed 3-D DDA method has better execution time, 5 times faster with 6 threads used in comparison with sequential methods.	Peng, et al., (2020). [17]
18	To implement parallel algorithms for Back Substitution, Conjugate Gradient and Gauss Seidel using OpenMP.	Comparative study of all the three algorithms using parallel and serial methods is done.	Paliwal, et al., (2022). [18]
19	To identify the computational cost of parallel and sequential methods to calculate vehicle forces and brake airflow while applying braking to prevent train accident prevention.	Parallel implementation provides an 80% reduction in model solution time.	Teodoro, et al., (2020). [19]
20	To study, Maximum-Shift string matching algorithm implementation with multi-core parallel method using OpenMP.	Performance of Maximum-Shift algorithm is altered with different data types of processing data, and it is studied with English text data and DNA database set.	AbdulRazzaq, et al., (2021). [20]
21	To develop Orthogonal Moments on block represented images.	Implementation done using OpenMP show significant performance for calculating the Orthogonal Moments using Image Block Representation (IBR).	Spiliotis, et al., (2021). [21]
22	To implement parallel and distributed computing Odd even transaction sorting algorithm.	Test results show Raspberry Pi clusters provide high performance speed.	Myint, et al., (2020). [22]
23	To implement K- means clustering algorithm in parallel mode using MPI.	Performance K- means clustering technique is compared with sequential and parallel execution.	Ragunthar, et al., (2021). [23]
24	To develop parallel scientific applications for the subject of physics and material science using OpenMP and MPI.	Designed four sequential programs of physics and material science subjects using parallel methods and verified its execution speed with respect to sequential method.	Aldinucci, et al., (2021). [24]

Literature review is conducted on the topic of SWOT(Strengths-Weaknesses-Opportunities-Threats) Analysis of different companies and listed in Table 2. This is done to know the method of performing SWOT Analysis for any specific company so that the knowledge of this could be used in analyzing the companies that are using Parallel Processing techniques.

Table 2: Scholarly literature review on SWOT Analysis of various Companies

S. No.	Area & Focus of the Research	Outcome of the Research	Reference
1	To develop a better SWOT model for Nvidia for analyzing a company's annual and financial document.	The result of the research shows that Nvidia is not an appropriate investment object for short-term analysis.	Zhao, M. (2023). [25]
2	To study Intel Corporations organization values with its crisis management.	It is found that organizational communication is an important aspect for change management and corporate culture integration for sustainable performance.	Kok, et al., (2023). [26]
3	To perform SWOT analysis of Samsung Electronics Company with internal and external analysis.	Provide a list of suggestions to the company to maintain sustainable success.	Hidiroglu, D. (2021). [27]
4	To perform the analysis of Dell's marketing mode using SWOT analysis method.	Relevant optimization suggestions are listed for Dell's IT industry.	Jiang, et al., (2022). [28]
5	To perform PEST and SWOT analysis to understand the marketing situation of Apple Inc.	Provided with the strategic marketing plans for the company's future development.	Chen, et al., (2021). [29]
6	To perform SWOT analysis of Dell Company.	Nine recommendations are given for future development.	Cheng, J. (2022). [30]
7	To perform SWOT analysis for Hydrogen fuel sustainable development in the GCC area.	Observed output of the research states that hydrogen-based economy in GCC regions depends on factors that are outbound.	Khan, et al., (2023). [31]
8	To identify the internal and external environment in Industry 4.0 manufacturing industry in India.	The result shows there is an increase in customers' trust while using online transactions. Also treat as employee resistance in embracing innovative tools and technologies.	Jain, et al., (2022). [32]
9	To conduct a survey on Egyptian HEIs to find the factor in ERP adoption and perform the SWOT analysis.	List of advantages in using ERP among Egyptian HEIs were found.	Soliman, et al., (2022). [33]
10	To perform the SWOT analysis on e-commerce tourism development strategy with big data management.	Identified the steps in taking development policy for scalable business.	Li, K. (2022). [34]

From the systematic literature review, it is found that four major Parallel Processing APIs used by various companies are - CUDA, OpenCL, OpenMP and MPI. There are research possibilities in performing SWOT(Strengths-Weaknesses-Opportunities-Threats) analysis of all four Parallel Processing APIs which are used by companies to develop Parallel Processing applications and systems. Research needs are there to identify major application domains of Parallel Processing, to perform company analysis of four major Parallel Processing APIs, to study about the companies which uses Parallel Processing APIs and list their major projects and to perform SWOT analysis of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI.

5. MAJOR APPLICATION AREAS OF PARALLEL PROCESSING :

Parallel Processing can be used in Computational Finance & Business Economics, Artificial Intelligence (AI), Machine Learning (ML), Data Science (DS), Numerical Analysis and Design of Algorithms where large amounts of data processing is a requirement.

(1) Computational Finance & Business Economics

Monte Carlo simulations are mainly adopted for implementing financial systems including product estimating, risk estimation, modelling of rate of interest etc. Monte Carlo models can be solved efficiently using Parallel Processing techniques. Stochastic programming methods like Decomposition and Interior Point Methods (IPMs), can be implemented effectively with parallel programming techniques.

(2) Artificial Intelligence (AI)

Parallel Processing methods are generally used in Artificial Intelligence due to its computational demands. Numerous areas of AI like Knowledge Representation, Image Processing, Logic and Production Rules, Natural Language Processing, Data Filtering and Data Mining uses parallel execution models. Robotic Process Automation (RPA) is the method to replace humans to perform repetitive office processes in a computer. As RPA tasks are highly repetitive, robots can be assigned tasks to perform in parallel to increase the usage of computing resources and execution efficiency.

(3) Machine Learning (ML)

Machine Learning is a set of procedures that offers models the capability to inherently acquire and progress from knowledge deprived of explicit programs. Performance of Machine Learning algorithms can be increased by parallel execution of common tasks like matrix multiplication that are used among numerous algorithms like classification, regression, and clustering techniques. In Neural Networks, there can be many features that are represented as parameters and to develop the model it requires huge volumes of data to train these parameters. These computationally rigorous processes take a higher time-complexity that can be optimized using parallel algorithms to train a deep neural network.

(4) Data Science (DS)

Parallel Processing techniques can be used widely in data science-based applications as it involves many computational tasks with enormous amounts of data. For example, in climate and weather science parallel programming methods can be largely used in both data analysis and computer modelling. Analysis of Seismic data sets can be done using numerical operators like Fourier transforms, integral transforms, convolutions, and other mathematical concepts. These operations can be performed in parallel to increase the system performance.

(5) Numerical Analysis

Parallel Numerical Analysis consists of using parallel techniques in solving numerical analysis problems like: - nonlinear equation, optimization, numerical integration, and ordinary differential equation. Parallel implementation of linear algebra computations, simultaneous exploration of different regions using multiple start points can be done. In optimization parallel methods are useful for the evaluation of objective and constraint functions and their derivatives. Parallel implementation of Runge-Kutta method, Newton's methods etc. are useful in solving differential equations faster than its sequential implementation.

(6) Design of Algorithm

Parallel algorithms are designed to divide a problem into subproblems and solve each sub-problem to produce the intermediate result by considering the synchronized actions. The result of the problem is calculated by combining the outcomes of each sub-problem. Design of parallel algorithms involves steps like: - Dividing a computational problem into subtasks which can execute simultaneously, developing parallel algorithms, Analysis of computational granularity, Minimizing the cost of parallel algorithms and assigning parallel task execution.

6. COMPANY DETAILS OF PARALLEL PROCESSING API - CUDA, OpenCL, OpenMP and MPI :

Parallel Processing Application Programming Interface

Parallel Processing Application Programming Interface (API) refers to programming interfaces or libraries that allow developers to write code for parallel execution of tasks on multiple processors or cores, thereby increasing the overall processing speed and performance of a system. These APIs provide a way to manage threads, processes, and data across multiple cores or processors, making it easier for developers to take advantage of Parallel Processing. Majorly used Parallel Processing APIs by various companies are: -

- CUDA - Compute Unified Device Architecture
- OpenCL - Open Computing Language
- OpenMP - Open Multi-Processing
- MPI - Message Passing Interface

Table 3 shows company details of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI, with the details related to API developed year, founder member who developed the API, Current CEO of the company, number of employees in the company and the company URL. Company details who have developed these Parallel Processing APIs are mentioned below: -

(1) CUDA - Compute Unified Device Architecture
Company Developed: Nvidia Corporation
Nvidia Corporation is a computer-based establishment that develops, designs and manufactures Graphics Processing Units (GPUs).

(2) OpenCL - Open Computing Language
Original Developer: Apple Inc.
Company Developed: Khronos Group
Khronos Group is an amalgam of 170 companies that are open, which are into the development and maintenance of interoperability standards in 3D graphics, virtual and augmented reality, parallel computing, AI.

(3) OpenMP - Open Multi-Processing
Company Developed: OpenMP Architecture Review Board
OpenMP Architecture Review Board (or OpenMP ARB) is a cluster of computer hardware and software-based companies like IBM, Intel, AMD, and Oracle Corporation.

(4) MPI - Message Passing Interface
Company Developed: It is considered as the De Facto standard for the exchange of information between processes in parallel programs.
Message Passing Interface (MPI) was developed at Austria, in 1991 with a set of researchers started with the work related to parallel computing and execution model.

Table 3: Company Detail of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI

Parallel Processing API	Company Developed	Company Details				
		Started	Founder	Current CEO	Employees	Company URL
CUDA	Nvidia Corporation	1993	Jensen Huang, Curtis Priem and Chris Malachowsky	Jensen Huang	25,000+	https://www.nvidia.com/en-in/
OpenCL	Apple Inc. and	2000	Consortium of companies: Intel	Neil Trevett is the current President of Khronos Group	44+	https://www.khronos.org/

	Khronos Group		Corporation, Sun Microsystems etc.			
OpenMP	OpenMP Architecture Review Board	1997	By both Intel, and the U.S. Department of Energy	Mr. Michael Wong of IBM Canada is the CEO of the OpenMP Architecture Review Board	10+	https://www.openmp.org
MPI	De Facto standard for communication among processes that model a parallel program	1991	Jack Dongarra, Tony Hey, and David W. Walker	MPI Group Committee Members	-	https://www.mpi-forum.org/

7. COMPANIES USING PARALLEL PROCESSING API :

Parallel Processing can be used in Computational Finance & Business Economics, Artificial Intelligence (AI), Machine Learning (ML), Data Science (DS), Numerical Analysis and Design of Algorithms where large amounts of data processing is a requirement. Foremost companies that use the Parallel Processing APIs and various applications, systems, models, and projects that are developed using Parallel Processing techniques are listed below: -

(1) Nvidia Corporation

Nvidia Establishment is an international computer software and hardware company based in California, America. The company majorly designed Graphics Processing Units (GPUs) and Application Programming Interfaces (APIs) for parallel applications. It also developed System on a Chip Units (SoCs) for mobile computing and automotive market applications.

- NVIDIA is the creator of CUDA Parallel Processing API, and they use it extensively in their products and solutions. NVIDIA does many research projects using CUDA in the application field of deep learning, healthcare, autonomous vehicles, and robotic processes.
- NVIDIA uses OpenCL in designing software development tools like CUDA toolkit that allow it to develop cross-platform applications.
- OpenMP is used by NVIDIA in optimizing its GPUs performance.
- CUDA is developed using MPI that allows applications to run on distributed systems.

(2) International Business Machines Corporation (IBM), nicknamed Big Blue

IBM is a multinational knowledge corporation located in New York, America. It has offices in 175 countries around the globe. IBM provides services in all three major fields like computer hardware, middleware, and software in the technology domain. It has services from mainframe systems to nanotechnology. It is an organization that focuses highly on research in various fields of technology.

- IBM uses CUDA in various projects like medical image analysis, quantum computing, and predictive analytics. It is also used in designing enterprise solutions and high-performance computing clusters.
- OpenMP is used by IBM to optimize the processor performance and software applications, to increase the scalability of systems.
- IBM used MPI in developing the Blue Gene supercomputer which is highly scalable.

(3) Advanced Micro Devices, Inc (AMD)

Advanced Micro Devices, Inc. (AMD) is an electronic device development company based in California, America. It works majorly on semiconductor development like computer processors.

- CUDA is used by AMD to increase the processing capacity of its graphics processors and data centers. It is working with CUDA in the project domain like: - artificial intelligence and virtual reality.
- OpenCL is used in the development of AMD's 3D content creation software named Radeon ProRender. It is used in GPU acceleration in complex processes.
- OpenMP is used by AMD in optimizing the performance of its processors and increasing the scalability of software applications.

(4) Intel Corporation (Intel)

Intel Corporation is an international system-based company in California, America. Intel is the world's biggest semiconductor chip manufacturing enterprise.

- Intel used OpenCL in the development of its graphics cards and in designing applications in the domain of molecular dynamics simulations and quantum computing.
- Intel uses OpenMP to improve the performance and scalability of its processors. OpenMP software development kit is developed by Intel for the Intel architecture.
- Intel MPI is the proprietary MPI library developed by Intel for providing fast communication between nodes in distributed computing.

(5) Cray Inc. (Cray)

Cray Inc. is an American supercomputer firm centered in Washington which builds supercomputer techniques for data storage and analytics. It is the subsidiary of Hewlett Packard Enterprise.

- Cray uses OpenMP in designing business applications for scientific and technical computing.
- MPI is used in the development of HPC data analytics platform.

(6) Siemens

Siemens is a German multinational company based in Munich, Germany. It is the largest industrial manufacturing company and maker of medical diagnostics equipment in Europe. It has principal company divisions of Digital Industries, Smart Infrastructure, Mobility, Healthcare and Financial Services.

- CUDA is used in industrial automation and control systems by Siemens. It is also used in advanced medical imaging equipment.
- Simcenter CFD is the software developed by Siemens for Computational Fluid Dynamics using OpenCL to reduce the simulation time and increase the accuracy of results.

(7) Baidu, Inc

Baidu is a Chinese worldwide technology establishment specialized in artificial intelligence and based in Beijing.

- Baidu is developing projects in AI, speech and image recognition and Natural Language Processing (NLP) using CUDA.

(8) Hewlett Packard Enterprise (HPE)

Hewlett Packard Enterprise (HPE) is an international information technology business based in Texas, America that provides services in domain like servers, storage, networking, software and consulting.

- HPE uses CUDA to accelerate the servers of HPE and on the company’s research projects like genomics research, machine learning, and natural language processing.

(9) Samsung

Samsung Group is a multinational manufacturing firm based in Seoul, South Korea.

- CUDA is used by Samsung in its AI based research and development projects including NLP, computer vision, and autonomous vehicle technology.

(10) Dell Inc. (Dell)

Dell Inc. is an American computer-based company that manufactures, sells, maintains, and supports personal computers (PCs) and its associated requirements. The company manages supply chain and electronic commerce to sell servers, data storage devices, network switches, computer hardware equipment's, printers, etc.

- Dell uses CUDA to increase the computing power of its data center applications like: - seismic data analysis, financial modeling, and genome sequencing.

(11) Apple Inc. (Apple)

Apple Inc. is the world's biggest computer-based enterprise in terms of profit, based in California, America. Apple was started as Apple Computer Company on April 1, 1976, by Steve Wozniak, Steve Jobs, and Ronald Wayne to build and market Wozniak's Apple I personal computer.

- Apple uses OpenCL to enable optimized performance of computing devices and technologies with high performance in rendering, video encoding, and real-time editing.

(12) Adobe Inc. (Adobe)

Adobe Inc. is an international technology driven company based in California, America. The company has developed major software products like: - Adobe Photoshop image editing software, Adobe Illustrator vector-based illustration software, Adobe Acrobat Reader, and the Portable Document Format (PDF). The company developed software instruments for audio-visual design, creating, editing, and publishing.

- Adobe uses OpenCL for performance optimization in a lot of their applications. Adobe's Premiere Pro, After Effects, and Photoshop are some of the popular products that use OpenCL to accelerate tasks like video editing, color grading, and image manipulation.

(13) Blackmagic Design Pty Ltd. (Blackmagic Design)

Blackmagic Design Pty Ltd. is a digital movie-based firm and hardware producer based in Victoria, Australia. The company developed high performing digital cameras and video making software like DaVinci Resolve and Blackmagic Fusion.

- The firm used OpenCL for its DaVinci Resolve video editing software, for color alteration and audio creation and editing through GPU processing.

(14) MathWorks

MathWorks is an American corporation that developed widely used mathematical computing software like MATLAB and Simulink, that support data analysis and simulation.

- MathWorks uses OpenCL in MATLAB software, to enable high-performance computing on GPUs, to solve large and complex computational problems.

(15) Maxon Computer (Maxon)

Maxon is a software company that produces software solutions for content creators which is based in Germany. Following are the major products of Maxon: - 3D software and Cinema 4D (used for content creation), Red Giant and Redshift renderer tools (used for editing, motion design and filmmaking), ZBrush (used for digital sculpting and painting) and Cinebench (used to evaluate hardware performance).

- OpenCL is used in Maxon's Cinema 4D for rendering complex 3D scenes and to accelerate tasks like reflections, shadows, and lighting.

(16) Unity Software Inc. (Unity Technologies)

Unity Software Inc. is a video game development company based in San Francisco; America that developed a product named Unity (a licensed game engine).

- Unity uses OpenCL to improve performance of complex visual effects.

(17) Microsoft Corporation (Microsoft)

Microsoft Corporation is an international technology-based establishment based in Washington, America. Microsoft's software products are Windows operating systems, Microsoft Office suite, and the Edge web browsers. Microsoft was founded by Bill Gates and Paul Allen in April 1975. Company dominates the personal computer operating system market with MS-DOS, followed by Windows.

- Microsoft uses OpenMP to optimize the performance of its software applications through Parallel Processing of Windows operating systems. Microsoft also used OpenMP in the Visual Studio development environment.

(18) Fujitsu Limited (Fujitsu)

Fujitsu is an ICT equipment and services corporation based in Tokyo, Japan. Company produces personal and enterprise computing products, like x86, SPARC and mainframe compatible servers.

- Fujitsu uses OpenMP to optimize the processors performance and increase the scalability of its software applications.

(19) Lenovo Group Limited (Lenovo)

Lenovo is an American Chinese international computer-based company that majorly design, manufacture, and market personal computers and devices like desktop computers, laptops, tablet computers and smartphones. It also develops terminals, servers, main frame, data storage devices, IT management systems and applications, and smart TVs.

- Lenovo uses OpenMP in its high-performance computing applications used for scientific computing, and in its business applications.

(20) Lawrence Livermore National Laboratory (LLNL)

LLNL is an institution in the field of science and technology based in California, America that prominently works on research and innovation. It was established in 1952, and funded by the US Department of Energy, and managed by National Security. The institution is accountable for ensuring protection to the nation's nuclear weapons. It is also responsible for preventing arms of mass ruin usage and ensuring native land security. It must solve various countrywide major problems like energy and ecological demands, technical research and outreach and commercial effectiveness.

- LLNL used MPI in its CANDLE project, which is a cancer research project with large-scale simulations.

(21) Argonne National Laboratory (ANL)

ANL is a Govt. sponsored research and development firm based in Illinois, United States. It is funded by the United States Department of Energy and overseen by the University of Chicago.

- MPI is used by ANL, in its supercomputer named Mira to make it highly scalable.

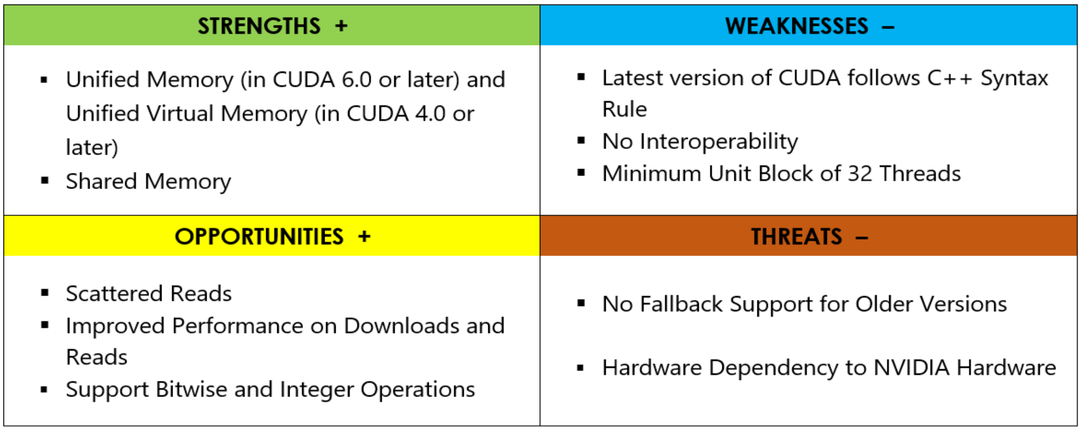
8. SWOT ANALYSIS OF PARALLEL PROCESSING API :

SWOT analysis is a management framework generally used to perform the company analysis by top level management as a critical analysis of a company's performance [25]. It is used in finding Strengths-Weaknesses-Opportunities-Threats of a particular business and associated policies. This paper, we have used SWOT analysis in identifying the various features of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI, along with the competency each one in the market of High-Performance Computing. Also, each APIs limitations and its performance in comparison with others is identified.

8.1. SWOT Analysis of CUDA Parallel Processing API:

SWOT Analysis of CUDA Parallel Processing API is shown in Table 4.

Table 4: SWOT Analysis of CUDA Parallel Processing API



STRENGTHS +	WEAKNESSES –
▪ Unified Memory (in CUDA 6.0 or later) and Unified Virtual Memory (in CUDA 4.0 or later) ▪ Shared Memory	▪ Latest version of CUDA follows C++ Syntax Rule ▪ No Interoperability ▪ Minimum Unit Block of 32 Threads
OPPORTUNITIES +	**THREATS –**
▪ Scattered Reads ▪ Improved Performance on Downloads and Reads ▪ Support Bitwise and Integer Operations	▪ No Fallback Support for Older Versions ▪ Hardware Dependency to NVIDIA Hardware

Source: [Author]

(1) Strengths

- Unified Memory (in CUDA 6.0 or later) and Unified Virtual Memory (in CUDA 4.0 or later)

  Unified Memory model represents single memory address-space for the entire processors in a computer, whereas Unified Virtual Memory model represents single virtual memory address-space for the whole memory in the computer, so that pointers can be accessed from GPU code. Unified Memory allows applications to allocate data by either CPUs or GPUs. Unified Memory provides higher performance than Unified Virtual Memory.

- Shared Memory

  Shared memory is assigned per thread group, so that all threads in the group have retrieve permission to the common shared memory. Shared memory is much faster than local and global memory access and threads retrieve details from shared memory by other threads of the same thread group. This leads to various uses like: - user-managed data caches, high-performance cooperative parallel algorithms, and global memory merging.

(2) Weaknesses

- Latest version of CUDA follows C++ Syntax Rule

  Newer versions of CUDA source code follow C++ language constructs, whereas earlier forms used C language rules. Thus modified CUDA programs may not be compatible with its older versions.

- No Interoperability

CUDA has very limited interoperability options, like one directional interoperability with OpenGL. OpenGL can retrieve details from CUDA memory, but CUDA cannot retrieve details from OpenGL memory.

- Minimum Unit Block of 32 Threads

  In CUDA, one thread block should have a minimum of 32 threads and a maximum of 1024 threads. If each of 32 threads yields the common running sequence, branches in the program code do not affect the performance.

(3) Opportunities

- Scattered Reads

  As Shared memory is given per thread group, all threads in the group have permission to retrieve the common shared memory. Thus, CUDA will be able to fetch data from any address in memory as Scattered Reads.
- Improved Performance on Downloads and Reads

  CUDA applications can improve their performance by reducing CPU-GPU data transfer.
- Support Bitwise and Integer Operations

  Single-precision floats like Integer division and modulo operations provide better performance with higher cost; and bitwise operations can be used to avoid the same. There are two types of runtime math operations with or without preceded underscores.

  Functions with __functionName() labeling standard represent precisely to the hardware level. They are high-speed but results in lower accuracy (e.g., __sinf(x) and __expf(x)). Functions with functionName() labeling standard are slower in execution but provide greater accuracy (e.g., sinf(x) and expf(x)). Execution of __sinf(x), __cosf(x), and__expf(x) are faster in running compared to sinf(x), cosf(x), and expf(x).

(4) Threats

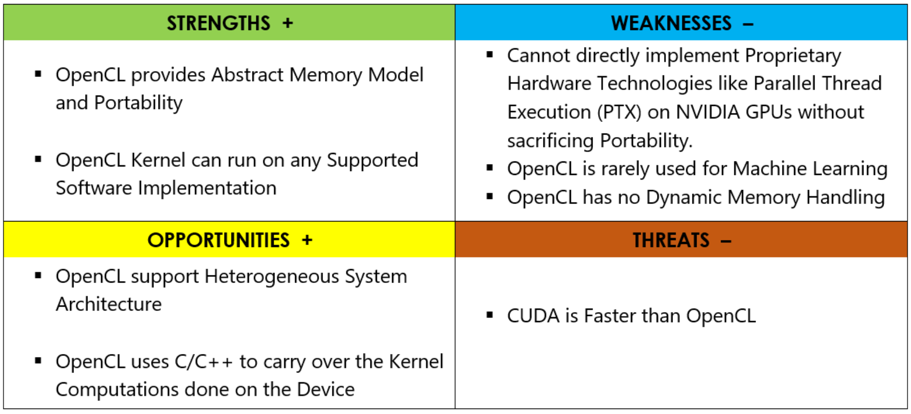
- No Fallback Support for Older Versions

  Later releases of CUDA do not provide alternative assistance for older versions.
- Hardware Dependency to NVIDIA Hardware

  CUDA is supported by only NVIDIA Hardware, thus it is highly machine dependent.

8.2. SWOT Analysis of OpenCL Parallel Processing API:

SWOT Analysis of OpenCL Parallel Processing API is shown in Table 5.**Table 5:** SWOT Analysis of OpenCL Parallel Processing API



STRENGTHS +	WEAKNESSES –
▪ OpenCL provides Abstract Memory Model and Portability ▪ OpenCL Kernel can run on any Supported Software Implementation	▪ Cannot directly implement Proprietary Hardware Technologies like Parallel Thread Execution (PTX) on NVIDIA GPUs without sacrificing Portability. ▪ OpenCL is rarely used for Machine Learning ▪ OpenCL has no Dynamic Memory Handling
OPPORTUNITIES +	**THREATS –**
▪ OpenCL support Heterogeneous System Architecture ▪ OpenCL uses C/C++ to carry over the Kernel Computations done on the Device	▪ CUDA is Faster than OpenCL

Source: [Author]

(1) Strengths

- OpenCL provides Abstract Memory Model and Portability

  OpenCL has an Abstract Memory Model that supports portability across various hardware devices. It also enables parallelism to the developers by acting as an interface between programmers and hardware. OpenCL API that supports cross-platform parallel program development on varied environments. OpenCL also supports intense memory requirements and data-parallel implementation.
- OpenCL Kernel, be able to execute on any supported Software Implementation

  As a framework, OpenCL allows developers to develop parallel programs which execute on both CPUs and GPUs of various hardware vendors like AMD, Intel, ATI, Nvidia etc.

(2) Weaknesses

- Cannot use Proprietary Hardware Technologies like Parallel Thread Execution (PTX) on NVIDIA GPUs deprived of Portability.
  Nvidia's CUDA programming supports Parallel Thread Execution (PTX) as a low-level parallel thread virtual machine and instruction set architecture for a thread execution from a parallel thread array. Cooperative Thread Array (CTA) is a group of threads which run a kernel concurrently or in parallel.
- OpenCL is rarely used for Machine Learning
  OpenCL has limited libraries for supporting Machine Learning. Qualcomm OpenCL accelerates Machine Learning operations, enables edge training, reduces CPU overhead of dispatching ML workloads and provides advanced math built-ins. oneDNN is an open-source cross-platform OpenCL extension for accelerating Deep Learning applications.
- OpenCL has no Dynamic Memory Handling
  As OpenCL does not have provision for Dynamic Memory Handling, it can either perform the creation of the largest possible array size for a particular program statically or create the dynamic array on CPU and pass it as a kernel argument.

(3) Opportunities

- OpenCL support Heterogeneous System Architecture
  OpenCL supports parallelism between Heterogeneous Computing Architectures. Communication between GPU and the processor is performed through C++; std::atomic template. The std::atomic template represents a type of atomic so that one thread can write to an atomic object while another thread can read from it.
- OpenCL uses C/C++ to carry over the Kernel Computations done on the Device
  OpenCL views a processing unit as a group of compute components consisting of work items. Each work item is a thread, and a group of threads is known as a computer component. OpenCL uses C/C++ to perform the kernel computations on the device like creating buffers, calling kernels, mapping the memory back to CPU, etc. OpenCL uses optimization techniques to improve the parallel computations done on GPU which results in better performance.

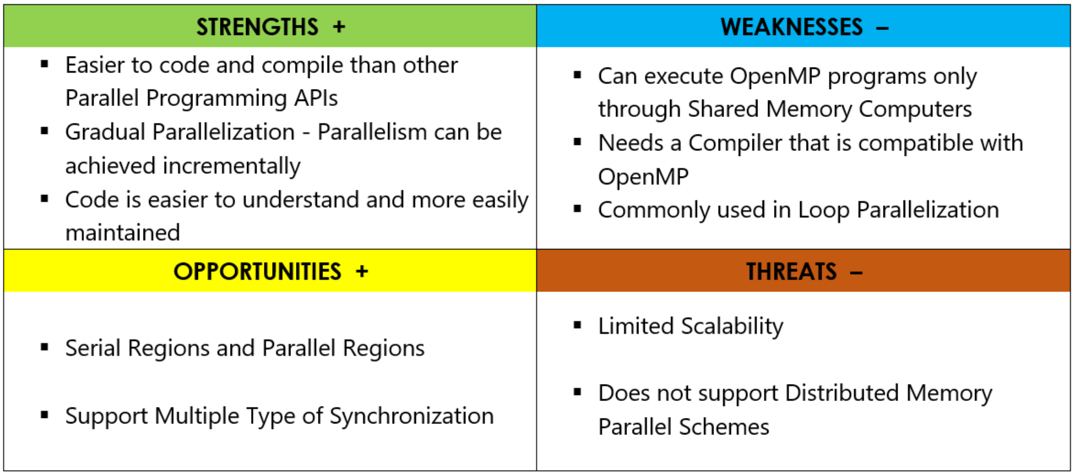
(4) Threats

- CUDA is Faster than OpenCL
  Execution time comparison of CUDA programs with OpenCL on NVIDIA GPUs shows that CUDA is thirty percentage speedier in comparison to OpenCL. OpenCL kernel can be compiled at runtime for achieving the portability that adds up to the OpenCL's running time.

8.3. SWOT Analysis of OpenMP Parallel Processing API:

SWOT Analysis of OpenMP Parallel Processing API is shown in Table 6.

Table 6: SWOT Analysis of OpenMP Parallel Processing API



STRENGTHS +	WEAKNESSES –
▪ Easier to code and compile than other Parallel Programming APIs ▪ Gradual Parallelization - Parallelism can be achieved incrementally ▪ Code is easier to understand and more easily maintained	▪ Can execute OpenMP programs only through Shared Memory Computers ▪ Needs a Compiler that is compatible with OpenMP ▪ Commonly used in Loop Parallelization
OPPORTUNITIES +	**THREATS –**
▪ Serial Regions and Parallel Regions ▪ Support Multiple Type of Synchronization	▪ Limited Scalability ▪ Does not support Distributed Memory Parallel Schemes

Source: [Author]

(1) Strengths

- Easier to code and compile than other Parallel Programming APIs
  OpenMP programs follow a structured programming style that makes it easier to understand. It is easier to locate the entry and exit of synchronized blocks of OpenMP programs.
- Gradual Parallelization - Parallelism can be achieved incrementally.
  The two principles on which OpenMP is designed are: - Sequential Equivalence and Incremental Parallelism. Sequentially Equivalent programs provide the same results even if it executes on one thread or many threads. These programs are easier to understand, write, and maintain. Incremental Parallelism is the process of converting serial code into a parallel code through incremental steps. At each increment, code can be verified for its correctness. OpenMP supports Incremental Parallelism for the shared memory model, known as Gradual Parallelization.
- Code is easier to understand and more easily maintained.
  OpenMP is a portable and scalable programming tool that provides developers an environment for designing parallel models for systems like standard personal system or high-performance supercomputer. OpenMP programs execution starts with one process known as master thread. The master thread executes sequentially until the first parallel section is met. When a parallel section is met, master thread performs: -
  a) Using the FORK command, create a set of threads.
  b) Becomes the master of the group of threads that are created and assigns thread id 0 within the group.

(2) Weaknesses

- Can execute OpenMP programs only through Shared Memory Computers
  OpenMP API is useful in creating multi-threaded, shared memory parallelism. One shared memory process contains many threads, that all threads perceive a shared variable similarly. It is the responsibility of the developer to ensure that the variable is updated by different threads as needed.
- Needs a Compiler that is compatible with OpenMP
  OpenMP parallelism uses compiler directives which are embedded in the source code. Compiler generated automatic parallelization is not supported by OpenMP.
- Commonly used in Loop Parallelization
  OpenMP supports parallelizing loops with few parameters for achieving the Parallel Processing of tasks. Loop Parallelization in OpenMP is known as a job-sharing model which can be placed inside a parallel section.

(3) Opportunities

- Serial Regions and Parallel Regions
  OpenMP allows a programmer to discrete a program into serial sections and parallel sections. This allows the development of a normal program as serial code, without any modification in OpenMP.
- Support Multiple Type of Synchronization
  When several threads are in execution concurrently, there is a requirement for thread synchronization so that one thread synchronizes with another thread at parallel regions. OpenMP has two classes of synchronization named: - explicit and implicit synchronization.

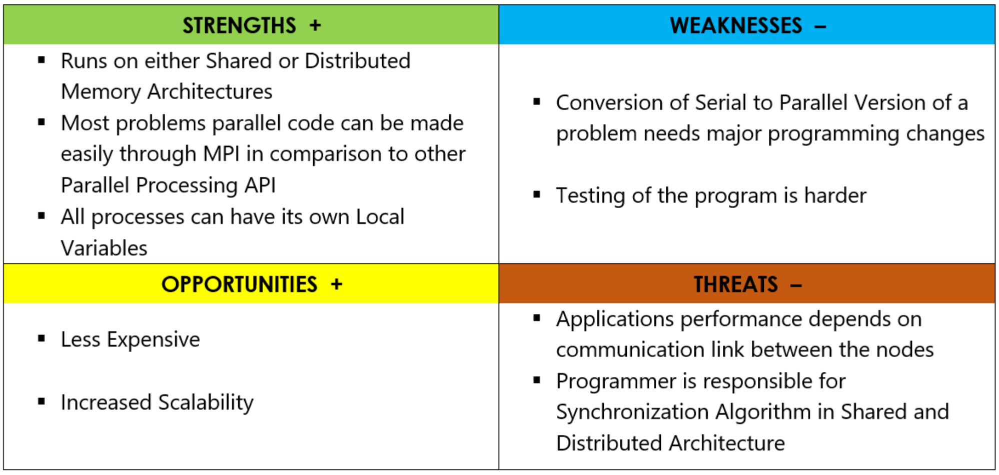
(4) Threats

- Limited Scalability
  Scalability of OpenMP programs depends on the performance of the shared memory node on which the program is executing.
- Does not support Distributed Memory Parallel Schemes
  Distributed memory systems represent computing models with multi processors, and each processor with its own memory. In Distributed Memory Parallel Systems programming jobs efficiently operate with local details, but for remote details job threads must communicate with remote processors to transmit details. OpenMP does not have the concept of communication with multiple processors for the exchange of remote data.

8.4. SWOT Analysis of MPI Parallel Processing API:

SWOT Analysis of MPI Parallel Processing API is shown in Table 7.

Table 7: SWOT Analysis of MPI Parallel Processing API



STRENGTHS +	WEAKNESSES –
▪ Runs on either Shared or Distributed Memory Architectures ▪ Most problems parallel code can be made easily through MPI in comparison to other Parallel Processing API ▪ All processes can have its own Local Variables	▪ Conversion of Serial to Parallel Version of a problem needs major programming changes ▪ Testing of the program is harder
OPPORTUNITIES +	**THREATS –**
▪ Less Expensive ▪ Increased Scalability	▪ Applications performance depends on communication link between the nodes ▪ Programmer is responsible for Synchronization Algorithm in Shared and Distributed Architecture

Source: [Author]

(1) Strengths

- Runs on either Shared or Distributed Memory Architectures

  Message Passing Interface (MPI) is the method used to transfer messages among multiple computers executing a parallel program in the distributed memory architecture. MPI also supports shared memory programming architecture, so that parallel programming on clusters of shared memory nodes can be done.

- Most problems parallel code can be made easily through MPI in comparison to other Parallel Processing API

  MPI allows the designing and implementation of programs that can run through parallel and distributed computing with coordination among multiple processes. MPI has more flexible control structures than other Parallel Processing APIs, which makes it a choice in solving an extensive range of problems.

- All processes can have its own Local Variables

  Every child process created in MPI has its own local variables, which makes it easier to execute parallel processes and share the results among multiple processes through communication.

(2) Weaknesses

- Conversion of Serial to Parallel Version of a problem needs major programming changes

  Designing an algorithm using parallel computing has two regions, regions of programs which cannot run in parallel (known as Serial Region) and regions of programs which can run in parallel (known as Parallel Region). MPI takes more time in designing Parallel Region in comparison with Serial Region.

- Testing of the program is harder

  General debugging tools support debugging by considering each process at a time that is appropriate for serial execution of a program. MPI programs must use a debugger that supports the execution model to see the execution result of multiple processes running in parallel.

(3) Opportunities

- Less Expensive

  MPI supports parallel programs on distributed computing architecture. Distributed systems are less expensive in comparison with multiprocessing systems with shared memory access. Thus, the use of MPI in developing parallel programs is more feasible in comparison with other APIs.

- Increased Scalability

  MPI supports distributed computing architecture where several systems can interconnect with each other in solving a parallel program, which increases the scalability of the application.

(4) Threats

- Applications performance depends on communication link between the nodes

  As MPI is used in parallel computing of programs in distributed environments, intra-node communication efficiency determines the application performance.

- Programmer is responsible for Synchronization Algorithm in Shared and Distributed Architecture Synchronization is the major design measure for parallel programming on shared and distributed architecture. Ensuring the synchronization in both shared and distributed architecture lies on the programmer who designs the parallel algorithms.

9. SUGGESTION AND RECOMMENDATIONS :

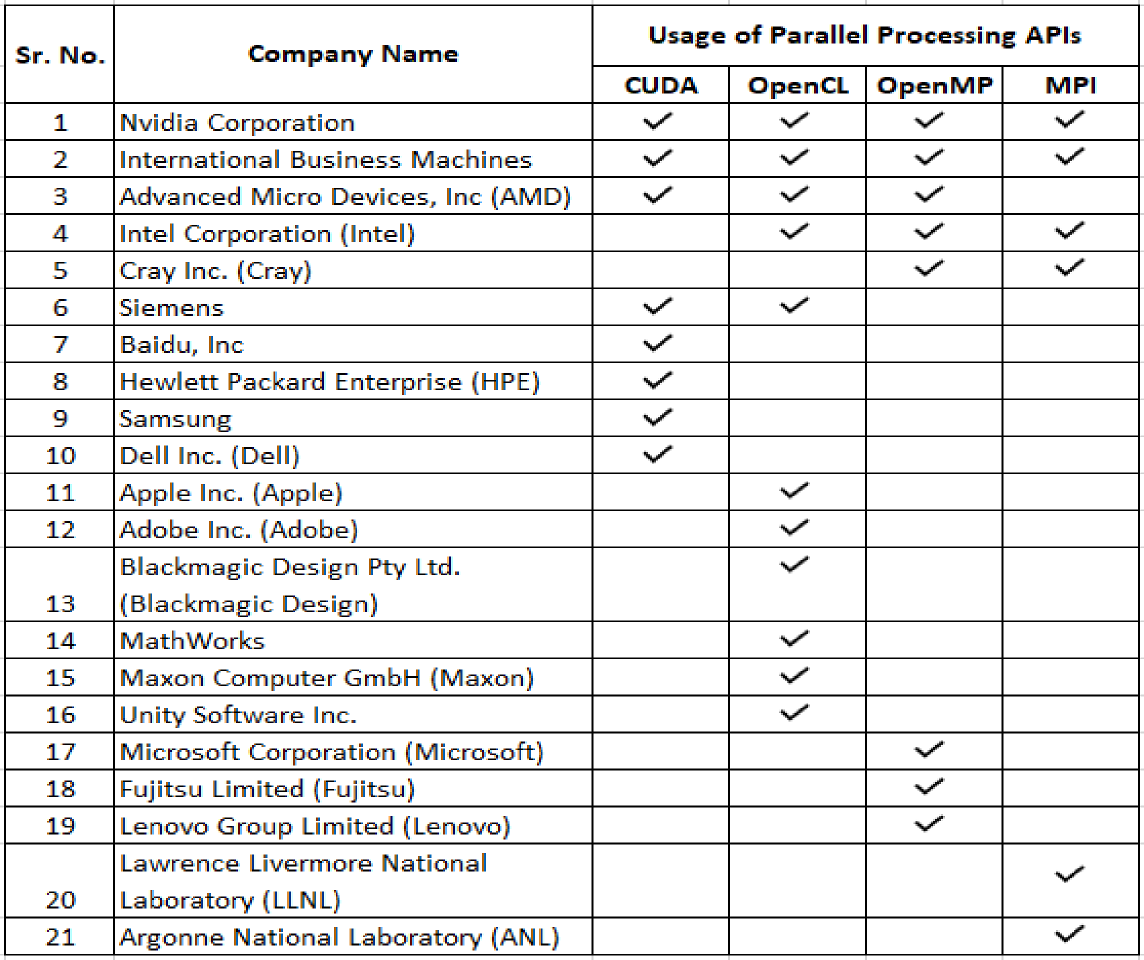
Based on the Company Analysis of Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI, following suggestions are proposed as recommendations:

(1) Parallel Processing techniques are used in various application areas like: - Monte Carlo simulations, Stochastic Programming Problems, Knowledge Representation, Image Processing, Logic and Production Rules, Natural Language Processing (NLP), Data Filtering, Data Mining, Robotic Process Automation (RPA), Machine Learning (ML) Algorithms and Numerical & Data Analysis.

(2) Four major Parallel Processing Application Programming Interfaces used by various companies are: - CUDA, OpenCL, OpenMP and MPI. These APIs are developed by the below mentioned companies: -
- CUDA - Compute Unified Device Architecture is developed by Nvidia Corporation.
- OpenCL - Open Computing Language is developed by Apple Inc. and Khronos Group.
- OpenMP - Open Multi-Processing is developed by OpenMP Architecture Review Board.
- MPI - Message Passing Interface is developed as De Facto standard for communication among processes that model a parallel program.

(3) Parallel Processing APIs CUDA, OpenCL, OpenMP and MPI are used by various companies to develop parallel applications and projects. Companies that are using CUDA, OpenCL, OpenMP and MPI techniques are shown in Table 8.

Table 8: Companies using Parallel Processing APIs CUDA, OpenCL, OpenMP and MPI



Sr. No.	Company Name	Usage of Parallel Processing APIs			
		CUDA	OpenCL	OpenMP	MPI
1	Nvidia Corporation	✓	✓	✓	✓
2	International Business Machines	✓	✓	✓	✓
3	Advanced Micro Devices, Inc (AMD)	✓	✓	✓	
4	Intel Corporation (Intel)		✓	✓	✓
5	Cray Inc. (Cray)			✓	✓
6	Siemens	✓	✓		
7	Baidu, Inc	✓			
8	Hewlett Packard Enterprise (HPE)	✓			
9	Samsung	✓			
10	Dell Inc. (Dell)	✓			
11	Apple Inc. (Apple)		✓		
12	Adobe Inc. (Adobe)		✓		
13	Blackmagic Design Pty Ltd. (Blackmagic Design)		✓		
14	MathWorks		✓		
15	Maxon Computer GmbH (Maxon)		✓		
16	Unity Software Inc.		✓		
17	Microsoft Corporation (Microsoft)			✓	
18	Fujitsu Limited (Fujitsu)			✓	
19	Lenovo Group Limited (Lenovo)			✓	
20	Lawrence Livermore National Laboratory (LLNL)				✓
21	Argonne National Laboratory (ANL)				✓

Source: [Author]

(4) SWOT analysis performed on CUDA and OPenCL shows that CUDA has hardware dependency to NVIDIA hardware, whereas CUDA is faster in execution in comparison with OpenCL.
(5) SWOT analysis performed on OpenMP and MPI shows that OpenMP has limited Scalability, whereas MPI can be used in both shared and distributed architectures.

10. CONCLUSION :

Parallel Processing techniques are extensively used in the application domain of Computational Finance & Business Economics, Artificial Intelligence, Machine Learning, Data Science, Numerical Analysis and Design of Algorithms. Company analysis of four major Parallel Processing APIs - CUDA, OpenCL, OpenMP and MPI shows that these APIs are recent techniques with a wide range of applications and possibilities. Various companies that are using these APIs show that these APIs have significant levels of application processing capabilities in the Parallel Processing domain and many new application areas can use the scope of these APIs further. SWOT analysis performed on these APIs will allow any researcher working in the Parallel Processing environment to select the appropriate APIs as per the research problem.

REFERENCES :

[1] Oh, B. W. (2019). Parallel algorithm for spatial data mining using CUDA. *Journal of Advanced Information Technology and Convergence*, *9*(2), 89-97. Google Scholar↗

[2] Afzal, A., Ansari, Z., & Ramis, M. K. (2020). Parallel performance analysis of coupled heat and fluid flow in parallel plate channel using CUDA. *Computational and Applied Mathematics*, *39*(1), 1-25. Google Scholar↗

[3] Afif, M., Said, Y., & Atri, M. (2020). Computer vision algorithms acceleration using graphic processors NVIDIA CUDA. *Cluster Computing*, *23*(1), 3335-3347. Google Scholar↗

[4] Imankulov, T., Daribayev, B., & Mukhambetzhanov, S. (2021). Comparative analysis of parallel algorithms for solving oil recovery problem using CUDA and OpenCL. *International Journal of Nonlinear Analysis and Applications*, *12*(1), 351-364. Google Scholar↗

[5] Wu, X., Wang, K., Li, Y., Liu, K., & Huang, B. (2020). Accelerating HAZE removal algorithm using CUDA. *Remote Sensing*, *13*(1), 85-112. Google Scholar↗

[6] Kim, M., & Baek, N. (2021). A 3D graphics rendering pipeline implementation based on the OpenCL massively Parallel Processing. *The Journal of Supercomputing*, *77*(1), 7351-7367. Google Scholar↗

[7] Shin, W., Yoo, K. H., & Baek, N. (2020). Large-scale data computing performance comparisons on SYCL heterogeneous Parallel Processing layer implementations. *Applied Sciences*, *10*(5), 1656 -1665. Google Scholar↗

[8] Ma, C., Jia, J., Liu, Z., Zhang, K., Huang, J., & Wang, X. (2022). Simulation of three-dimensional phase field model with LBM method using OpenCL. *The Journal of Supercomputing*, *78*(8), 11092-11110. Google Scholar↗

[9] Mahony, A. O., Zeidan, G., Hanzon, B., & Popovici, E. (2022). A parallel and pipelined implementation of a Pascal-simplex based multi-asset option pricer on FPGA using OpenCL. *Microprocessors and Microsystems*, *90*(1), 104508-104526. CrossRef↗

[10] Yamato, Y. (2020). Study of Parallel Processing area extraction and data transfer number reduction for automatic GPU offloading of IoT applications. *Journal of Intelligent Information Systems*, *54*(1), 567-584. Google Scholar↗

[11] You, L., Yang, E., & Wang, G. (2020). A novel parallel image encryption algorithm based on hybrid chaotic maps with OpenCL implementation. *Soft Computing*, *24*(1), 12413-12427. Google Scholar↗

[12] Abdelaziz, M., & Moradzadeh, M. (2019). Monte-Carlo simulation based multi-objective optimum allocation of renewable distributed generation using OpenCL. *Electric Power Systems Research*, *170*(1), 81-91. CrossRef↗

[13] Afzal, A., Ansari, Z., & Ramis, M. K. (2020). Parallelization of numerical conjugate heat transfer analysis in parallel plate channel using OpenMP. *Arabian Journal for Science and Engineering*, *45*(1), 8981-8997. Google Scholar↗

[14] Dongarra, J., Gates, M., Haidar, A., Kurzak, J., Luszczek, P., Wu, P., & Relton, S. D. (2019). PLASMA: Parallel linear algebra software for multicore using OpenMP. *ACM Transactions on Mathematical Software (TOMS)*, *45*(2), 1-35. CrossRef↗

[15] Spiliotis, I. M., Bekakos, M. P., & Boutalis, Y. S. (2020). Parallel implementation of the image block representation using OpenMP. *Journal of Parallel and Distributed Computing*, *137*(1), 134-147. CrossRef↗

[16] Ketchaya, S., & Rattanatranurak, A. (2023). Parallel Multi-Deque Partition Dual-Deque Merge sorting algorithm using OpenMP. *Scientific Reports*, *13*(1), 6408-6424. Google Scholar↗

[17] Peng, X., Chen, G., Yu, P., Zhang, Y., Zhang, H., & Guo, L. (2020). A full-stage parallel architecture of three-dimensional discontinuous deformation analysis using OpenMP. *Computers and Geotechnics*, *118*(1), 103346-103351. CrossRef↗

[18] Paliwal, M., Chilla, R. R., Prasanth, N. N., Goundar, S., & Raja, S. P. (2022). Parallel implementation of solving linear equations using OpenMP. *International Journal of Information Technology*, *14*(3), 1677-1687. Google Scholar↗

[19] Teodoro, Í. P., Eckert, J. J., Lopes, P. F., Martins, T. S., & Santos, A. A. (2020). Parallel simulation of railway pneumatic brake using OpenMP. *International Journal of Rail Transportation*, *8*(2), 180-194. CrossRef↗

[20] AbdulRazzaq, A. A., Hamad, Q. S., & Taha, A. M. (2021). Parallel implementation of maximum-shift algorithm using OpenMP. *Indonesian Journal of Electrical Engineering and Computer Science*, *22*(3), 1529-1539. Google Scholar↗

[21] Spiliotis, I. M., Sitaridis, C., & Bekakos, M. P. (2021). Parallel Computation of Discrete Orthogonal Moment on Block Represented Images Using OpenMP. *International Journal of Parallel Programming*, *49*(1), 440-462. Google Scholar↗

[22] Myint, K. N., Zaw, M. H., & Aung, W. T. (2020). Parallel and distributed computing using MPI on raspberry Pi cluster. *International Journal of Future Computer and Communication*, *9*(1), 18-22. Google Scholar↗

[23] Ragunthar, T., Ashok, P., Gopinath, N., & Subashini, M. (2021). A strong reinforcement parallel implementation of K-means algorithm using Message Passing Interface. *Materials Today: Proceedings*, *46*(1), 3799-3802. CrossRef↗

[24] Aldinucci, M., Cesare, V., Colonnelli, I., Martinelli, A. R., Mittone, G., Cantalupo, B., & Drocco, M. (2021). Practical parallelization of scientific applications with OpenMP, OpenACC and MPI. *Journal of parallel and distributed computing*, *157*(1), 13-29. CrossRef↗

[25] Zhao, M. (2023). NVIDIA's Investment Feasibility and Weighted SWOT Model. *Highlights in Business, Economics and Management*, *3*(1), 227-236. Google Scholar↗

[26] Kok, S. L., & Siripipatthanakul, S. (2023). Change Management Model in Corporate Culture and Values: A Case Study of Intel Cooperation. *Advance Knowledge for Executives*, *2*(1), 1-30. Google Scholar↗

[27] Hidiroglu, D. (2021). The Strategies in Global Semiconductor Industry: The SWOT Analysts of Samsung Electronics Company. *American Journal of Management*, *21*(2), 107-118. Google Scholar↗

[28] Jiang, Z., Lin, H., Jiang, M., & Ling, W. (2022). Analysis of Dell's Marketing Strategy. *Law and Economy*, *1*(1), 29-37. Google Scholar↗

[29] Chen, X., Liu, Y., & Gong, H. (2021). Apple Inc. Strategic Marketing Analysis and Evaluation. *International Conference on Economic Management and Cultural Industry (ICEMCI 2021), 3*(1), 3053-3061. Google Scholar↗

[30] Cheng, J. (2022). Sustainable Competitive Advantage Analysis and Recommendations for Dell Company to Achieve Success. *Open Journal of Social Sciences*, *10*(12), 486-497. Google Scholar↗

[31] Khan, M. I., & Al-Ghamdi, S. G. (2023). Hydrogen economy for sustainable development in GCC countries: A SWOT analysis considering current situation, challenges, and prospects. *International Journal of Hydrogen Energy*, *48*(28), 10315-10344. CrossRef↗

[32] Jain, V., Ajmera, P., & Davim, J. P. (2022). SWOT analysis of Industry 4.0 variables using AHP methodology and structural equation modelling. *Benchmarking: An International Journal*, *29*(7), 2147-2176. Google Scholar↗

[33] Soliman, M., & Noorliza, K. (2022). Adopting enterprise resource planning (ERP) in higher education: A SWOT analysis. *International Journal of Management in Education, 16*(1), 20-39. CrossRef↗

[34] Li, K. (2022). SWOT analysis of e-commerce development of rural tourism farmers' professional cooperatives in the era of big data. *IET Communications*, *16*(5), 592-603. CrossRef↗

****