FUTURE EDUCATION

Future Institute of Technology

NEWSLETTER

2022-2023

Message from Head of the Department

Prof. (Dr.) Soumadip Ghosh
M.Tech,Ph.D

I would like to take this opportunity to welcome you to the Department of Computer Science & Engineering, Future Institute of Technology, Kolkata, India.

The Computer Science and Engineering fields are in constant evolution. So our knowledge needs to be updated to keep us up-to-date. That's why we have given importance to the core subjects as well as the advanced subjects such as Data Science, Cloud Computing, Artificial Intelligence (AI), Machine Learning (AI), Deep Learning (DL), Web Technologies, Internet of Things (IoT), and Android app development.

We believe that people who have a thorough understanding of the core and advanced subjects of the discipline will be able to quickly adapt to rapid technological advancements.

We want our students' education to serve as the foundation for a lifetime of learning.

Ever since we started our journey way back in 2014, our department has produced hundreds of professionals and has established a name for itself in the country and abroad. They have consistently excelled in the highly competitive industrial environment at top-ranking companies.

I attribute this success to the winning combination of a dedicated faculty team that works hard at imparting quality education to our students.

Learning is a continuous process and does not end with the acquisition of a degree, especially because steady and rapid advances in computing technologies shorten the life of tools and techniques prevalent today.

As a result, we do not try to turn our students into walking manuals for any one language or software. Instead, students are given a solid foundation in computer science, taught how to solve problems, and made flexible.

We think that this method of teaching and learning, along with the acquired practical experience, will be successful. We prepare our students to meet the challenges presented by the software industry during their Industrial Training at reputable firms.

Applications

Technical Report: Advancing Algorithms with Accelerated Mathematics

Abstarct::

Mathematics is the foundation of many algorithms in computer science and various fields. This technical report explores the symbiotic relationship between algorithms and accelerated mathematics. We delve into the impact of faster math computations on algorithm performance, highlighting the advancements and benefits achieved through improved mathematical techniques. The report showcases real-world examples and discusses the implications for future algorithm development.

1. Introduction:

Algorithms are essential tools that solve complex problems in numerous domains, from scientific simulations to data analysis and machine learning. The efficiency of algorithms often hinges on the mathematical computations they rely upon. Faster math computations not only improve algorithm performance but also open new avenues for solving previously intractable problems.

2. Importance of Faster Math in Algorithms:

Faster math computations, achieved through optimized numerical methods and hardware acceleration, lead to several crucial benefits for algorithms:

a. Reduced Computational Complexity: Accelerated math techniques can transform computationally intensive algorithms into more efficient ones by reducing the number of operations required.

b. Real-time Processing: Algorithms that involve real-time decision-making or control systems benefit from faster math, enabling timely responses to changing inputs.

c. Scalability: Faster math allows algorithms to handle larger datasets or more complex models without sacrificing execution speed.

d. Enabling New Algorithms: Faster math makes it possible to develop algorithms for problems that were previously impractical due to computational limitations.

3. Advancements in Faster Math Techniques:

The field of accelerated mathematics has made significant strides, contributing to better algorithms across various domains:

a. Parallel Computing: Leveraging multi-core processors, GPUs, and specialized hardware accelerators has enabled algorithms to perform parallel computations, leading to substantial speedup.

b. Approximation Algorithms: Faster math techniques often involve approximations that maintain accuracy while reducing computation time, benefiting optimization and simulation tasks.

c. Numerical Libraries: Improved numerical libraries and optimized linear algebra routines enhance the performance of algorithms by utilizing faster math computations.

d. Quantum Computing: Quantum computing's ability to solve certain mathematical problems exponentially faster holds promise for revolutionizing algorithm design in the future.

4. Case Studies:

Real-world examples demonstrate the impact of faster math on algorithm performance:

a. Image Processing: Faster Fourier Transforms and convolution operations have revolutionized image processing algorithms, enabling real-time applications like video streaming and computer vision.

b. Machine Learning: Accelerated linear algebra and matrix operations have expedited training and inference in machine learning algorithms, enabling rapid model iteration and deployment.

c. Scientific Simulations: Faster numerical solvers have enhanced the accuracy and efficiency of simulations in fields such as fluid dynamics, enabling researchers to explore complex scenarios with unprecedented detail.

5. Future Implications:

The advancements in accelerated mathematics have profound implications for the future of algorithm development:

a. Algorithmic Innovation: Faster math opens doors to innovative algorithms that were previously too slow to be practical.

b. Real-time Analytics: Algorithms can make critical decisions in real-time, enabling applications in autonomous systems, financial trading, and healthcare monitoring.

c. Edge Computing: Faster math enables more sophisticated processing at the edge, enhancing the capabilities of edge devices in the Internet of Things (IoT) ecosystem.

d. Interdisciplinary Synergy: Collaborations between mathematicians and algorithm developers will lead to new breakthroughs that leverage both mathematical theory and accelerated computation.

6. Conclusion:

The intersection of faster math and algorithm development is a dynamic realm that continually reshapes the landscape of computation. As mathematical techniques evolve and hardware accelerators advance, algorithms stand to benefit from increased efficiency, scalability, and novel possibilities. Embracing accelerated mathematics is essential for driving the next generation of algorithms that power our digital world.

-Subham Chakroborty,asst prof.

Microservices with Java Spring Boot: Current Scenario in 2023

A large application is made up of many small, independent services known as "microservices," which connect with one another across a network or cloud.

Every service in this architecture is a standalone functional unit that can be created, tested, and implemented independently of the others. Compared to a monolithic architecture, where all the functionality is contained in a single, sizable codebase, this enables greater flexibility and scalability.

Due to its loose coupling, modifications to one service should not have an impact on the other service. The program is now simpler to scale, update, and maintain. Large, sophisticated systems that need to manage a lot of traffic and be scaled horizontally are best suited for microservices architecture.

A highly accepted Java framework for creating Restful web services and microservices is Spring Boot. With the help of Spring Boot, developers can quickly build Spring-based applications without having to repeatedly write the same boilerplate settings.

For developers to create common design patterns and address numerous infrastructure issues in Microservices projects while concentrating on their primary business issues, Spring Cloud offers a variety of tools or modules.

Spring Boot and Spring Cloud have emerged as the de facto guidelines for creating microservices architecture in the Java community. The following figure illustrates this.

Figure 1: Java Spring Boot microservices architecture

Key elements of a Java Spring Boot microservices architecture consist of:

1. Core Services: Every service is a standalone functional unit that can be created, tested, and implemented separately from the others.

2. Service registry: All of the services in the system, together with their locations and capabilities, are listed in a database called a service register. It enables services to find and connect with one another.

3. API Gateway: A single entry point for all incoming queries to the microservices is an API gateway. It serves as a reverse proxy, forwarding requests to the proper service and managing operations like rate limitation and authentication.

4. Message bus: A message bus is a messaging system that enables asynchronous communication between services. Protocols like RabbitMQ, HTTP, or Kafka can be used for this.

5. Monitoring and logging: In order to track the health of the services and solve issues, monitoring and logging are essential.

6. Service discovery and load balancing: This component is in charge of finding service instances and balancing traffic to the best service instances based on load and availability.

7. Continuous integration and continuous deployment: It is advised to utilise a tool like Jenkins, TravisCI, or CircleCI to automate the process of creating, testing, and deploying microservices in order to make the development and deployment of microservices as seamless as possible.

In 2022–'23, the Spring team published Spring Boot 3 which comes with improved @ConstructorBinding, enhanced Log4j2, native image support for GraalVM, an updated Micrometre, and a few other new features.

It is also crucial to be familiar with the new features and enhancements added in this most recent version of the Spring Framework because Spring Boot 3 requires *Spring Framework 6*.

-kamalesh karmakar,asst prof.