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.

Cybersecurity's Effect In 2022–2023

SOME GENERAL TRENDS AND PREDICTIONS WERE ANTICIPATED FOR THIS TIME FRAME BASED ON HISTORICAL PATTERNS.

Ransomware Keeps Changing: Threat actors were predicted to use cutting-edge techniques to breach systems and demand a ransom from businesses as ransomware attacks grew more sophisticated and targeted. It would probably be more difficult for victims to recover their data without paying the ransom if encryption and other obfuscation techniques were used.

AI and Machine Learning in Cybersecurity: It was anticipated that more cybersecurity tools and defense systems would incorporate AI and machine learning. These tools can help with threat detection and mitigation as well as vulnerability analysis for security professionals looking at huge amounts of data.

Smart device and IoT security issues: It was anticipated that worries about smart devices' security vulnerabilities would increase as the Internet of Things (IoT) and other connected devices continue to expand. Insecure firmware, a lack of encryption, and weak passwords were a few of the problems that could result in widespread attacks

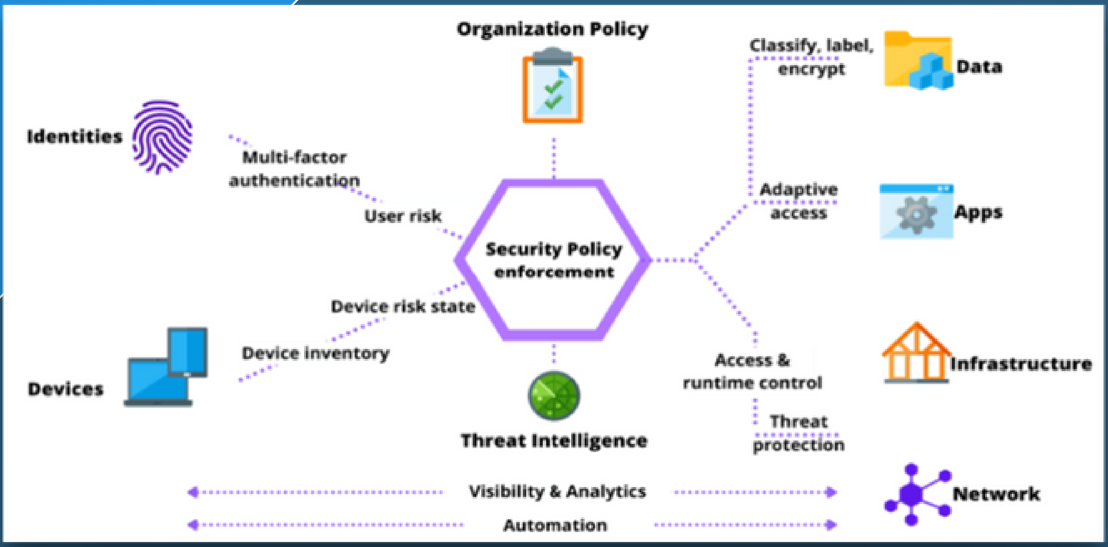
Challenges with cloud security: Increasingly, businesses are moving their infrastructure and services to the cloud, making it more important than ever to protect those environments. If not properly addressed, errors, insider threats, and data breaches might increase in frequency.

Supply Chain Attacks: To gain access to target organizations, cybercriminals were likely to take advantage of supply chain vulnerabilities. Attacks on suppliers and third-party vendors could serve as a gateway for attacks on more significant, high-value targets.

Lack of Qualified Personnel to Handle Cybersecurity Challenges: It was predicted that the demand for skilled cybersecurity professionals would exceed the workforce that was available.

Adoption of zero-trust architecture: As a security model to prevent unauthorized access, the idea of zero trust, which holds that no user or device should be automatically trusted within a network, was likely to gain attraction.

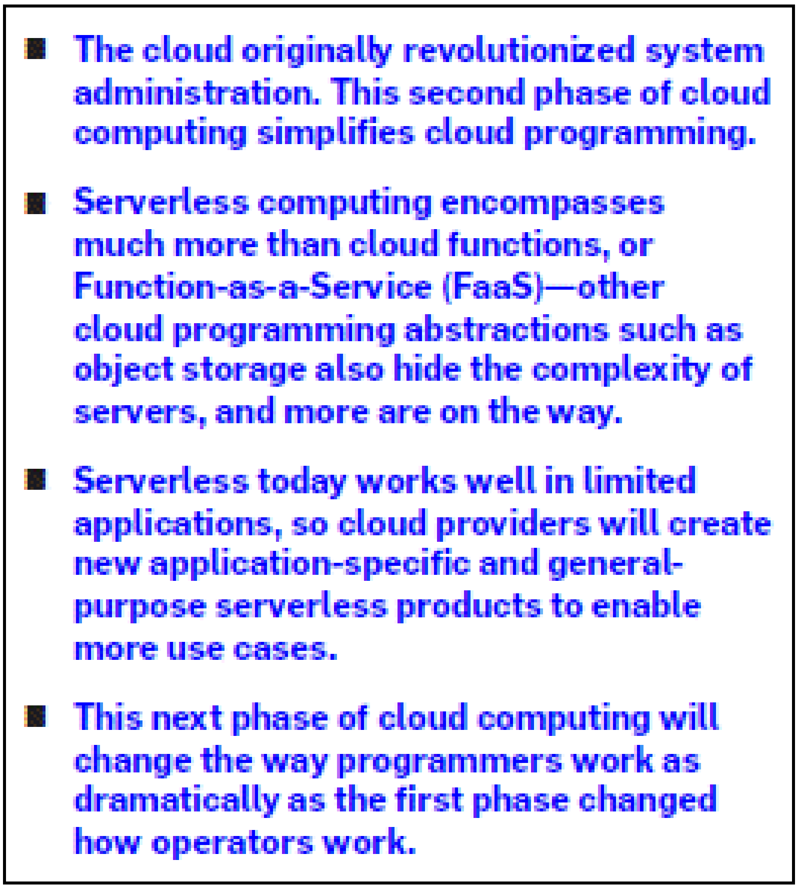
Regulation and Compliance: It was anticipated that governments and regulatory bodies around the world would focus more on cybersecurity, resulting in stricter compliance requirements for companies and organizations.



Prof. (Dr.) Soumadip Ghosh

What Serverless Computing is and should become: The Next Phase of Cloud Computing

We know that cloud computing provided the illusion of infinitely scalable remote servers without charging a premium for scale, as renting 1,000 servers for one hour costs the same as renting one server for 1,000 hours, and that economies of scale for the cloud provider allowed it to be surprisingly inexpensive.



- The cloud originally revolutionized system administration. This second phase of cloud computing simplifies cloud programming.
- Serverless computing encompasses much more than cloud functions, or Function-as-a-Service (FaaS)—other cloud programming abstractions such as object storage also hide the complexity of servers, and more are on the way.
- Serverless today works well in limited applications, so cloud providers will create new application-specific and general-purpose serverless products to enable more use cases.
- This next phase of cloud computing will change the way programmers work as dramatically as the first phase changed how operators work.

To emphasize the change of focus from servers to applications, this new phase has become known as *serverless computing,* although remote servers are still the invisible bedrock that powers it. In this article, we call the traditional first phase *serverful computing.*

Analogy: To attend a remote conference, you either rent a car or hail a taxicab to get from the airport to your hotel. Car rental is like serverful computing, where you must wait in line, sign a contract, reserve the car for your whole stay no matter how much you use it, drive the car yourself, navigate to the hotel, pay for parking, and fill it with fuel before returning it. The taxi is like serverless computing, where you simply need to give the hotel name and pay for the ride; the taxi service provides a trained driver who navigates, charges for the ride, and fills the gas tank. Taxis simplify transportation, as you don't need to know how to operate a car to get to the hotel. Moreover, taxis get higher utilization than rental cars, which lowers costs for the taxi company. Depending on the length of the conference, the cost of car rental, the cost of parking, the cost of gas, and so on, taxis are not only easier, they might even be cheaper.

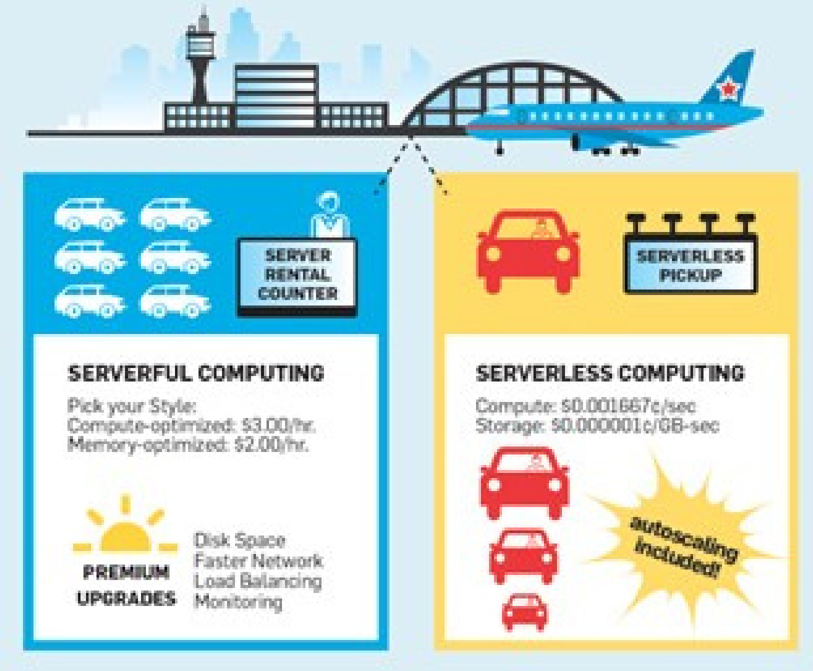


Figure 1. Cloud computing approaches compared to rides from an airport: Serverful as renting a car and serverless as taking a taxi ride.

Understanding What Serverless Is Today

Cloud functions capture much of the mindshare in serverless computing, but they are one of many services in the serverless cloud. The excitement around FaaS is well justified because it offers a glimpse of what general-purpose serverless computing might look like, yet BaaS services comprise a much larger, and older, set of serverless services.

For example, AWS initially offered their S3 object storage as a remote backup and archival service, years before announcing EC2 virtual machine rental. You can think of S3 as a precursor to serverless computing that offered "diskless storage," that is, providing storage but hiding the disks. Over time, cloud providers offered additional BaaS services to help serverful computing. Message queues (for example, AWS SQS, Google Cloud Pub/Sub) were another early service. Later came key-value databases (for example, Google Cloud Datastore, AWS DynamoDB, Azure CosmosDB) and SQL-based big data query engines (for example, AWS Athena, Google BigQuery).

When AWS Lambda launched in 2015 it was the first cloud functions product and offered something unique and compelling: the ability to execute nearly any code that runs on a server. It included support for several programming languages and for arbitrary libraries, all on a pay-as-you-go basis, operating securely and at any scale. However, it imposed certain limitations on the programming model that even today restrict it to certain applications. These include a maximum execution time, the lack of persistent state, and restricted networking.

Today, several serverless environments can run arbitrary code, each catering to a particular use case. For example, Google Cloud Dataflow and AWS Glue allow programmers to execute arbitrary code as a stage in a data processing pipeline, while Google App Engine can be thought of as a serverless environment for building Web applications.
These many serverless offerings have in common the three essential qualities of serverless computing: an abstraction that hides the servers, a pay-as-you-go cost model, and excellent autoscaling. Taken together they offer a set of alternatives that may be combined to meet an ever-growing range of applications.

The Next Phase of Cloud Computing

Perhaps the best way to understand the shift that serverless computing represents is to focus on the first of the essential qualities (as noted previously): providing an abstraction that hides servers and thus simplifies the programming and operating model. From the outset, cloud computing provided a simplified operating model, but simplified programming comes from hiding servers. The future evolution of serverless computing, and in our view of cloud computing, will be guided by efforts to provide abstractions that simplify cloud programming

It is striking how little cloud computing has changed how programmers work to date, especially when compared to the impact it has had on operators. Much of the software that runs in the cloud is the exact same software that runs in a traditional data center. Compare the programming skills most in demand today against those needed 10 years ago and you will notice that the core skill set has changed very little, even as specific technologies come and go. By contrast, the operator's job has changed tremendously. Installing and maintaining servers, storage, and networks are largely things of the past replaced by a focus on managing virtualized infrastructure through cloud provider APIs, and by the DevOps movement, which emphasizes the technical and organizational aspects of change management.

What makes programming the cloud hard? While it is possible to use the cloud with just one server, this offers neither fault tolerance nor scalability nor pay-as-you-go, so most cloud programming quickly becomes distributed systems programming. When writing distributed systems, programmers must reason about the data center's spatial extent, its various partial failure modes, and all of its security threats. In the language of Fred P. Brooks, these concerns represent "accidental complexity," which arises from the implementation environment and stands in contrast to "essential complexity," which is inherent in the functionality that the application provides. At the time of Brooks's writing, high-level languages were displacing assembly language, freeing programmers from reasoning about complex machine details such as register allocation or data layout in memory. Just as high-level languages hide many details of how a CPU operates, serverless computing hides many details of what it takes to build a reliable, scalable, and secure distributed system.

We next consider alternative approaches to serverless abstraction, including ones that exist today and ones that we imagine. These vie to answer the question, "if not servers, then what?" We group these alternative abstraction approaches into application-specific and general-purpose categories. Application-specific abstractions solve a particular use case, and several of them exist in products today. General-purpose abstractions must work well in a broad variety of uses and remain a research challenge.

-Payel Pramanik
asst prof.