Serverless computing in financial services

High performance computing—with more speed, accuracy and flexibility

Authors

Michael Behrendt is a Distinguished Engineer in the IBM Cloud development organization. He is responsible for IBM's technical strategy around serverless & Function-as-a-Service.

In that context, he's also the chief architect for the IBM serverless offering, IBM Cloud Functions. Before that, he was the chief architect of the core platform of IBM Bluemix and was one of the initial founding members incubating it.

Michael has been working on cloud computing for more than 13 years and has 35 patents. He is located in the IBM Research & Development Laboratory in Boeblingen, Germany.

Neil Cowit is the Worldwide Cloud Offering Manager for High Performance Computing at IBM. HPC has been a significant part of Neil's 20-plus-year career within the financial services technology sector. He has held leadership and individual contributor roles in Product Management, Development, Sales and Marketing.

Contents

Introduction

If Mrs. Wallis Simpson were alive today and involved in financial services, she may very well have modified her famous quote from “You can never be too rich or too thin.” to “You can never have too much compute capacity.”

There are numerous situations that exponentially increase the computational capacity financial firms must provide on a daily basis. They must deal with competitive pressures, regulatory reporting, security, fast-moving market swings, and innovative but complicated financial products.

This situation is made even more complex given everyone is already awash in data that needs to take into account international data regulations with different requirements for handling and transparency. Adding to these issues is the desire and need for intra-day risk reporting, analysis, optimization and operating transparency.

Let’s face it, speed and accuracy can’t be separated. One without the other is useless. The person with the right answer first, wins. The wrong answer can seriously harm a firm’s reputation, financial stability or even lead to incarceration of company officers. Chief technology officers (CTOs) constantly struggle with the issue of how to provide sufficient controlled compute capacity that’s auditable, secure and cost-effective.

So how do you win when it takes several hundred thousand or more computations to get to the correct answer especially when you often need that answer in a few seconds—at worst—in a few minutes and no firm has unlimited compute capacity on-premises.

Wouldn’t it be terrific if we only had to solve static or deterministic models as opposed to stochastic or probabilistic financial models? Imagine being able to cover all the possible contingencies in proportion to their likelihood.

Monte-Carlo simulations, in their various forms, aren’t new to financial services and make up the backbone of a wide range of risk, regulatory and optimization models.

The accuracy of a Monte-Carlo simulation is proportionate to the square root of the number of factors in the model. In other words, the greater the number of factors, the more accurate the results will be.

Regulatory agencies across the financial services markets are demanding a higher degree of certainty of outcome from the models. This higher degree of accuracy comes with a penalty—it requires more time or more resources—or both. To achieve the desired accuracy, the model needs more data and significantly more iterations and computations.

Grid computing, or high-performance computing, whether located on-premises or in the cloud, continues to deliver the performance levels these embarrassingly repetitive, parallel and computationally intensive calculations require.

The questions are:

- Can serverless computing add any value to this critical component of survival in the financial markets?
- Can serverless computing provide the near-instantaneous results, security, auditable results and dynamic flexibility that financial institutions require?

Whether you need to meet the regulatory reporting requirements of FRTB, price yield curves, derivatives or need to shorten the execution time of deep learning or machine learning models, financial institutions are using serverless computing, or as IBM® calls it, IBM Cloud™ Functions.

The goal is being able to model uncertainty by simulating its' seemingly endless variables, as quickly and accurately as possible. All you need is unlimited compute capacity, all the time in the world, and some funding.

This white paper provides guidance for deploying a Monte-Carlo simulation with IBM Cloud Functions.

Benefits of serverless computing

- Operations are quickly scalable
- Resources are more cost effective and flexible
- Multiple, independent, complex calculations complete faster
- Developers build and deploy more quickly

What issues can serverless computing help address?

As previously stated, improving model outcome requires inclusion of as many of the variants that drive model precision as possible, given available time and compute capacity.

Serverless computing, in many instances, is the next logical step in the evolution of a category of high-performance computing in the cloud for Financial Services—consisting of three, key attributes:

- **No infrastructure management**
  As the term "serverless" implies, it's about not having to deal with the management of servers. All infrastructure management is abstracted away from the developer, who can now focus on providing the actual business logic. Monitoring, patching, problem management, capacity management as well as the underlying operating system and infrastructure of the runtimes executing the code, is taken care of by the provider. Obviously, this method also speeds time to market.

- **Transparent scaling**
  The serverless infrastructure always scales the available capacity to fit exactly the current needs of the running functions. This capability results in never being over- or under-provisioned and delivering stable response times to users.

- **Never pay for idle time**
  As a result of the previous point, the customer never has to pay for idle capacity. If no load needs to be processed, no computer processing unit (CPU), memory or disk capacity is allocated, so no costs are being generated. As load increases or decreases, only the capacity that's exactly needed is allocated and therefore, needs to be paid for.

Technical perspective

In a serverless computing system, the technical unit of execution is usually a container. The typical case for today's serverless implementations is that one container runs one process that executes the function code in a single thread—which always processes one request or event at a time.

Since a container always has a dedicated amount of memory and CPU assigned to it, each processing step can rely on having dedicated CPU and memory capacity available. If a function is invoked many times in parallel, the serverless engine takes care of provisioning the corresponding number of containers to serve those requests or take them from a pre-warmed pool.

For example, if 1,000 requests arrive in parallel, the serverless engine would spin up 1,000 containers, each of them handling one request. This execution model lends itself well to running virtually any kind of embarrassingly parallel CPU- or memory-intensive workload—Monte-Carlo simulations being one of them.

IBM Cloud Functions: Implementing serverless computing

IBM Cloud Functions is a service offered on the IBM Cloud that implements the serverless behavior previously described. It's available in various regions around the world. The functions or actions processed by IBM Cloud Functions can be implemented in essentially any programming language, including node, Java®, Python, Go, Ruby, server-side Swift and more.

As data-hungry functions are registered, the amount of memory to be made available to each individual function invocation must be specified. Once a function is registered, it can be invoked using a Hypertext Transfer Protocol (HTTP) call or an event source, such as a database, firing events whenever a new record is created.

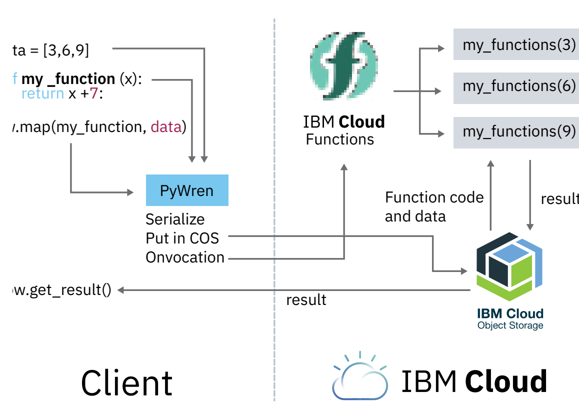

Figure 1. Functional diagram of IBM Cloud Functions.

Benefits of IBM Cloud Functions

This architecture allows executing embarrassingly parallel workloads—including Monte-Carlo simulations—in a very simple manner:

- The computational logic is captured in an action.
- Written in a language of choice.
- The user provides the input data relevant for driving the Monte-Carlo simulation.
- Store it somewhere, for example in IBM Cloud Object Storage.
- The user would then execute the invocation of the action as many times as needed, potentially thousands of times in parallel.
- Once all parallel computations are finished and the results are stored in a central location, the user can act on those results or perform further analysis.

With this approach, the user benefits from all serverless attributes previously described—not having to be concerned about infrastructure. The user gets transparent scaling and only pays exactly for the compute resources utilized.

How do IBM Cloud Functions differ from other providers' serverless functions? Here are the key differences:

- IBM Cloud Functions is the only serverless computing offering from a large vendor that's based on an independent, open source project. It's based on Apache OpenWhisk and provides customers the flexibility to run their serverless computing workloads on IBM Cloud Functions—and any other place they choose.
- The open source codebase is the exact same code as that used for IBM Cloud Functions. Clients get all the hardening, performance optimization, and so on that IBM and the open source community put into the project.
- IBM Cloud Functions provides a maximum degree of flexibility regarding which workloads are running in a serverless mode. Besides a series of natively supported languages, IBM Cloud Functions also allows you to package workloads in a Docker container and run the workload in that container.
- For many workloads, IBM Cloud Functions is the cost-competitive choice. Clients can be sure they're not spending more than is absolutely needed to run a given workload—while getting a rich set of features.

Conclusion

The possibilities for serverless computing reach far beyond that of Monte-Carlo simulations discussed in this brief paper. Microservices, API backends, mobile backends, data-in-motion and at-rest processing, chatbots and more are all improved through the proper use of IBM Functions. There are many ways to speed deep and machine learning efforts and serverless design can have an extraordinary effect on shortening these efforts.

IBM Cloud Functions and serverless computing in general, is a technology that is evolving at a tremendous pace. The speed with which new or modified offerings can be brought to market has a direct impact on the bottom line.

Remember, cloud is a methodology,
not a destination.

Always learn, plan, develop, deploy, test, execute and start again.

IBM is driving this and many other technologies to advance the overall realm of computing, whether it resides on-premises, hybrid or cloud native.

IBM looks forward to working with you and encourages you to contact us with any questions or comments.

Glossary

Basel Committee on Banking Supervision (BCBS)
The Basel Committee on Banking Supervision (BCBS) is the primary global standard setter for the prudential regulation of banks and provides a forum for regular cooperation on banking supervisory matters. Its 45 members comprise central banks and bank supervisors from 28 jurisdictions.

www.bis.org/bcbs

CCAR
The Comprehensive Capital Analysis and Review (CCAR) is an annual exercise by the Federal Reserve to assess whether the largest bank holding companies operating in the United States have sufficient capital to continue operations throughout times of economic and financial stress and that they have robust, forward-looking capital-planning processes that account for their unique risks.

www.federalreserve.gov/supervisionreg/stress-tests-capital-planning.htm

Container
A container is a standard unit of software that packages up code and all its dependencies, allowing applications to run quickly and reliably from one computing environment to another.

www.docker.com/resources/what-container

Dodd-Frank Act supervisory stress testing (DFast)
Dodd-Frank Act supervisory stress testing is a forward-looking quantitative evaluation of the impact of stressful economic and financial market conditions on financial institutions' capital.

This test serves to inform the Federal Reserve, financial institutions, and the general public about how financial institutions' capital ratios might change under a hypothetical set of stressful economic conditions.
All bank holding companies (BHC) with $100 billion or more in total consolidated assets and US intermediate holding companies (IHC) are currently subject to Dodd-Frank supervisory stress testing.

www.federalreserve.gov/supervisionreg/stress-tests-capital-planning.htm

Function (also referred to as functional event or function instance)
A function is a single-purpose block of code. The unit of computing was once a virtual machine. It evolved to a container. A function is the latest step in this evolution. A function can be as simple as a piece of code that performs a calculation on some data.

FRTB
Fundamental review of trading book (FRTB) is loosely defined as a set of proposals by BCBS as framework for the next generation market risk regulatory capital rules for large, internationally active banks.

www.quora.com/What-is-FRTB-fundamental-review-of-trading-book-how-will-it...

Hypervisor
A hypervisor is a process that separates a computer's operating system and applications from the underlying physical hardware. The hypervisor drives the concept of computer virtualization by allowing the physical (host) machine to operate multiple virtual machines (guests) to help maximize the effective use of computing resources such as memory, network bandwidth and CPU cycles.

www.networkworld.com/article/3243262/virtualization/what-is-a-hypervisor.html

High-performance computing (HPC)
High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly. The term applies especially to systems that function above a teraflop or 1012 floating-point operations per second.

searchdatacenter.techtarget.com/definition/high-performance-computing-HPC

MIFiD
The markets in financial instruments directive (MiFID) is a regulation that increases the transparency across the European Union's financial markets and standardizes the regulatory disclosures required for particular markets.

MIFiD | Investopedia
www.investopedia.com/terms/m/mifid.asp

Monte-Carlo simulation
Monte Carlo methods (or Monte Carlo experiments) are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. Their essential idea is using randomness to solve problems that might be deterministic in principle. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to use other approaches. Monte Carlo methods are mainly used in three problem classes:[1] optimization, numerical integration, and generating draws from a probability distribution.

en.wikipedia.org/wiki/Monte_Carlo_method

Monte-Carlo Simulation, or probability **simulation**, is a technique used to understand the impact of risk and uncertainty in financial, project management, cost, and other forecasting models. Uncertainty in Forecasting Models.

www.riskamp.com/files/RiskAMP%20-%20Monte%20Carlo%20Simulation.pdf

Open banking
See PSD2

OpenWhisk
Apache OpenWhisk is an open source, distributed serverless platform that executes functions in response to events at any scale.

openwhisk.apache.org

www.wired.co.uk/article/open-banking-cma-psd2-explained

Serverless computing
Serverless computing doesn't mean there are no servers. Serverless computing adds another layer of abstraction to a cloud computing infrastructure. Serverless computing (or function-as-a-service, FaaS) is an emerging application deployment architecture that completely hides server management from the users (that is, tenants).

FOR MORE INFORMATION

To learn more about the role of serverless computing in financial institutions, please contact your IBM representative or IBM Business Partner, or visit: **ibm.com**/cloud/functions

IBM Corporation
New Orchard Road
Armonk, NY 10504

49022849-USEN-00