

# ИСПОЛЬЗОВАНИЕ БЕССЕРВЕРНЫХ ВЫЧИСЛЕНИЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ ОБАРАБОТКИ ИНФОРМАЦИИ

Богдан А. А.

Кафедра программного обеспечения информационных технологий,  
Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь  
E-mail: alexeybogdan@icloud.com

*В данной работе будет дано понятие бессерверным вычислениям, будут рассмотрены их достоинства и недостатки. Приведены примеры существующих облачных решений, позволяющих разрабатывать приложения, используя бессерверные вычисления. Проанализирована разница стоимости равных по мощности бессерверных и серверных вычислений на облачном сервисе Amazon Web Services.*

## ВВЕДЕНИЕ

Задача обработки информации - это, в большинстве случаев, высоконагруженный процесс, который требует высокие мощности и соответственно дорогостоящее оборудование. В некоторых случаях процесс обработки информации выполняется по определенному расписанию либо сторонний сервис является триггером его запуска. Это значит что дорогостоящее оборудование большую часть времени работает вхолостую.

Напротив, для такого рода приложений можно использовать подход, когда не нужно покупать дорогостоящее оборудование, а использовать бессерверные вычисления. При их использовании, стоимость зависит от того, сколько раз процесс обработки информации был запущен.

В данной работе будут рассмотрены бессерверные вычисления, которые позволят сократить издержки в несколько раз.

## I. ПОНЯТИЕ БЕССЕРВЕРНЫХ ВЫЧИСЛЕНИЙ, ИХ ДОСТОИНСТВА И НЕДОСТАТКИ

Бессерверные вычисления (англ. serverless computing) — модель облачных вычислений, в которой платформа динамично руководит выделением машинных ресурсов. Иногда бессерверные вычисления также называют «Функция как услуга» (англ. Function as a Service, FaaS), потому что единицей кода является функция, которая выполняется платформой. По сути, для выполнения каждого запроса создается отдельный контейнер, который уничтожается после выполнения.

Конечно, бессерверные вычисления требуют аппаратного обеспечения. Это название используется потому, что пользователям такой платформы не надо иметь дело с настройкой серверов для запуска кода: все серверные настройки и планирование вычислительных ресурсов скрыты от пользователей и управляются платформой. Бессерверный код может быть частью приложений, построенных на традиционной архитектуре, например, на микросервисах.

Разрабатывая приложения, используя бессерверные вычисления, отсутствует необходи-

мость в обслуживании инфраструктуры, поскольку разработчики могут арендовать большую часть ресурсов, необходимых при разработке ПО. Более того, бессерверная архитектура имеет возможность масштабирования в соответствии с рабочей нагрузкой приложения. Разработчикам больше не нужно приобретать дополнительную инфраструктуру для обслуживания непредвиденного роста. Также, одним из достоинств является то, что нет необходимости писать сценарии для развертывания приложения на сервер. Для развертывания бессерверного приложения разработчикам просто нужно скомпилировать свой код, заархивировать его и загрузить на бессерверную платформу.

Бессерверные вычисления имеют и свои недостатки. Одним из них является ограничение ресурсов. Это необходимо для того, чтобы платформа могла справляться со скачками нагрузок и противостоять внешним атакам злоумышленников. Также существует риск небезопасной работы, так как разработчики запускают свои функции на общей платформе. Таким образом, при загрузке вредоносного кода возможны негативные последствия для работы других приложений.

Главным достоинством бессерверных вычислений является то, что это модель «плати, как ты используешь». Вам просто нужно заплатить за время выполнения вашей функции — продолжительность выполнения вашего кода и количество срабатываний. Сравнивая с традиционным подходом использования серверов, вам не нужно нести расходы за неиспользуемые и «незанятые» ресурсы.

## II. ПРОВАЙДЕРЫ БЕССЕРВЕРНЫХ ВЫЧИСЛЕНИЙ

Приведем примеры провайдеров бессерверных вычислений:

- Amazon Lambda Functions;
- IBM Cloud Functions;
- Microsoft Azure Functions;
- Google Cloud Functions.

Более подробно рассмотрим Amazon Lambda Functions. AWS Lambda - это управляемая событиями бессерверная вычислительная служба, которая позволяет запускать код без подготовки или управления серверами и может расширять другие службы AWS с помощью настраиваемой логики. Лямбда может автоматически запускаться в ответ на несколько событий, таких как HTTP-запросы через Amazon API Gateway, изменения данных в хранилище Amazon S3 или в базе данных Amazon DynamoDB, добавление нового элемента в очередь Amazon SQS. AWS Lambda запускает код в высокодоступной вычислительной инфраструктуре и выполняет все функции администрирования базовой платформы, включая обслуживание серверов и операционных систем, выделение ресурсов и автоматическое масштабирование, мониторинг кода и ведение журнала.

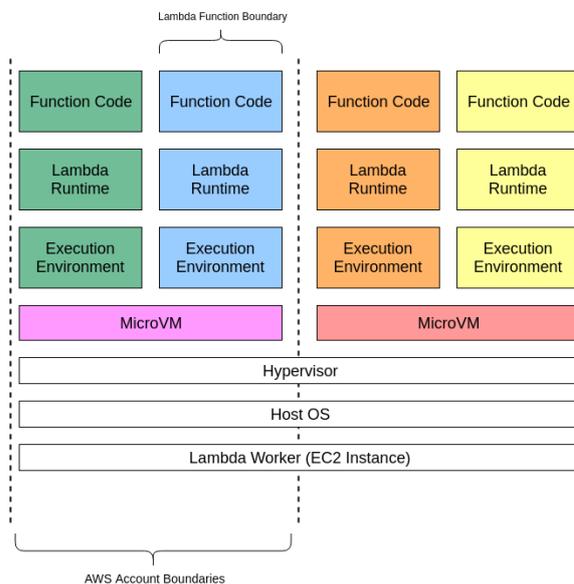


Рис. 1 – Принцип работы AWS Lambda

Рассмотрим принципы работы AWS Lambda. Каждая функция выполняется в одной или нескольких выделенных средах выполнения, которые используются в течение всего срока действия функции, а затем уничтожаются. Каждая среда выполнения содержит один параллельный вызов, но может повторно использоваться в нескольких последовательных вызовах одной и той же функции. Среда выполнения работают на аппаратных виртуальных машинах (microVM). MicroVM предназначен только для одной учетной записи AWS, но может быть повторно использован средами выполнения других функций учетной записи. MicroVM упакованы на аппаратной платформе, принадлежащей

AWS и управляемой ей (Lambda Workers). Среда выполнения никогда не передаются между функциями, а MicroVM никогда не распределяются между учетными записями AWS (см. рис 1).

### III. СРАВНЕНИЕ СТОИМОСТИ СЕРВЕРНЫХ И БЕССЕРВЕРНЫХ ВЫЧИСЛЕНИЙ

Ниже приведен пример оформления простой таблицы. Рассмотрим стоимости AWS EC2 (сервис, предоставляющий пользователям облачные сервера на операционных система Linux, Windows) и AWS Lambda для следующей задачи: необходимо анализировать покупки пользователей и определить популярную категорию товара в зависимости от региона. Время выполнения такой функции примем равным 10 секунд. В таблице 1 вы можете увидеть сравнительную характеристику стоимости равных по мощности бессерверных и серверных вычислений за месяц использования при различных условиях выполнения задачи.

Таблица 1 – Сравнительная характеристика

Кол-во запусков в час	СТОИМОСТИ	
	AWS Lambda, USD	AWS EC2, USD
1	0	14.976
6	0	14.976
10	0	14.976
20	0	14.976
30	0.59	14.976
60	7.75	14.976
120	22.2	14.976

Как вы можете заметить, если функция обработки информации выполняется несколько раз в час, то стоимость такой AWS Lambda функции будет бесплатна или стоить значительно дешевле, чем выделенный сервер. Выделенный сервер имеет смысл использовать, если средняя нагрузка на него достигает более чем тридцать процентов.

### СПИСОК ЛИТЕРАТУРЫ

1. Moving to Serverless with AWS Lambda [Electronic resource] / Paul Johnston. – O'Reilly. – Mode of access: <https://learning.oreilly.com/learning-paths/learning-path-moving/9781492054887>. – Date of access: 01.09.2019.
2. AWS Lambda Developer Guide [Electronic resource] / Amazon. – Amazon Web Services. – Mode of access: <https://docs.aws.amazon.com/lambda/latest/>. – Date of access: 01.02.2018.
3. In Depth AWS Lambda Overview [Electronic resource] / Georgi Velinov. – Medium. – Mode of access: <https://medium.com/faun/in-depth-aws-lambda-overview-1eeb4580696b>. – Date of access: 14.04.2019.