How Time & Space Complexity impacts your Infrastructure

Edwin Moedano Cardiel

Prácticas modernas para crear software con calidad y sabor

#SGVirtual

Edwin Moedano

- Cloud Senior Engineer @ Caylent
- Research and Development career
- MSC in Computer Science
- Music & Lyrics lover

Table of Contents

Our Session Goals!

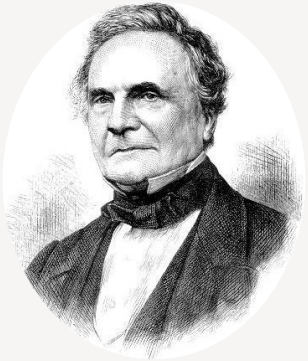
At the end of this session, you will be able to:

- Understand what is time and space complexity in algorithms
- Identify how time and space complexity impacts our infrastructure and budget
- Have a better understanding on how to choose technology based on time and space complexity.

Introduction - Algorithms

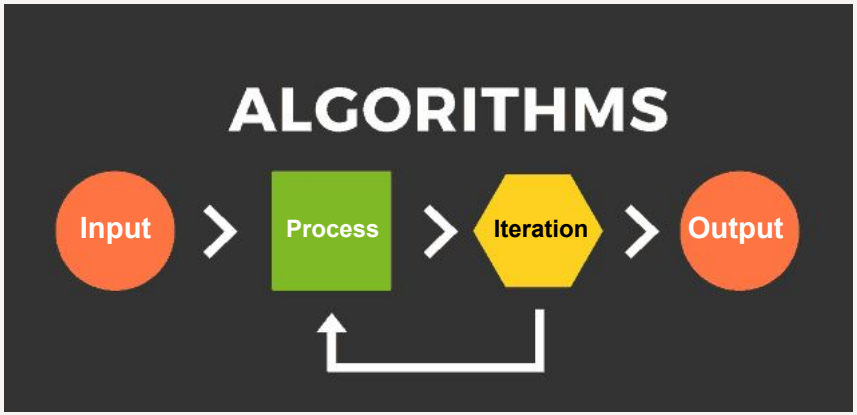
“Tan pronto como exista un motor analítico, necesariamente guiará el curso futuro de la ciencia. Siempre que se busca algún resultado con su ayuda, surge la pregunta: ¿mediante qué proceso de cálculo puede la máquina llegar a estos resultados en el menor tiempo?”

Charles Babbage



Introduction - Algorithms

We **aspires** to seek **efficient ways** to **solve** our daily **tasks** and the predominant thought **process behind** innovation and **technology** is to make **life easier** for people by **providing** ways to **solve problems** they may encounter.



Introduction - Algorithms

E**ncouraging** engineers to **write algorithms** that are **efficient**, **faster** and take up **less memory** to perform better.

Step #01: Start.

Step #02: Create two variables (a & b).

Step #03: Store integer values in ‘a’ and ‘b.’ -> Input

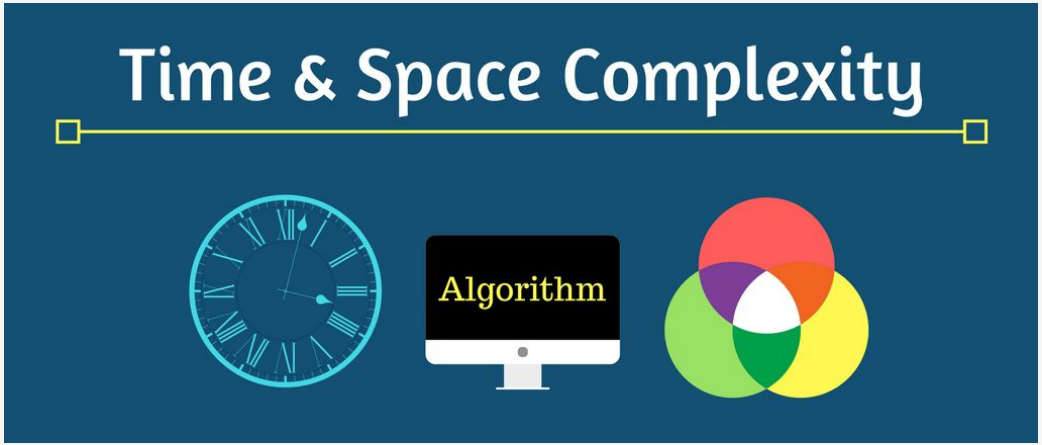
Step #04: Create a variable named ‘Sum.’

Step #05: Store the sum of ‘a’ and ‘b’ in a variable named ‘Sum’ -> Output

Step #06: End.

Introduction - Algorithms

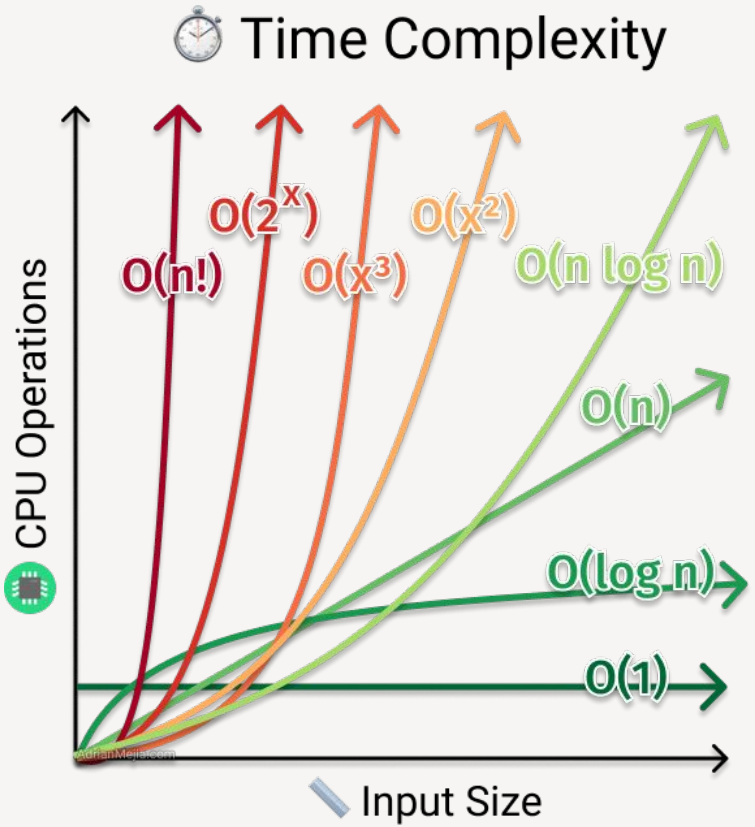
For those **requirements** we need to **take** a **deep breath** and **understand** how **time & space complexity** impacts the **performance** of our **applications** and **infrastructure**.



What is Time Complexity?

Time Complexity

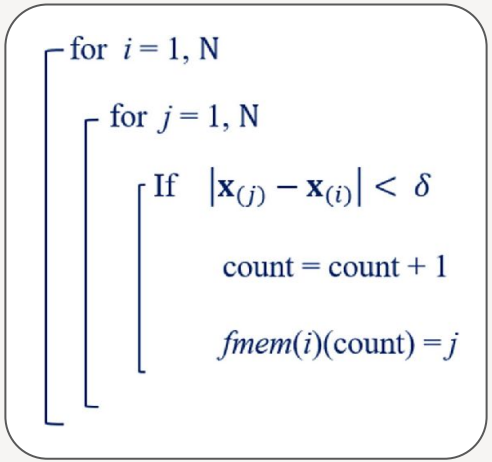
Is the **time taken** by the **algorithm** to **execute** each set of **instructions**. It is always better to **select** the most **efficient** algorithm when a simple problem can solve with different methods.



Time Complexity

Brute force: For many **non-trivial problems**, there is a **natural brute force** search **algorithm** that checks every possible solution.

- Typically takes **2N** time or worse $\mathbf{N^2}$ for inputs of size **N**.
- Unacceptable in practice.



Time Complexity

Depending on the algorithms we choose, complexity **increases** when **input size doubles**.

	n	$n \log_2 n$	n^2	n^3	1.5^n	2^n	$n!$
$n = 10$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	4 sec
$n = 30$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	< 1 sec	18 min	10^{25} years
$n = 50$	< 1 sec	< 1 sec	< 1 sec	< 1 sec	11 min	36 years	very long
$n = 100$	< 1 sec	< 1 sec	< 1 sec	1 sec	12,892 years	10^{17} years	very long
$n = 1{,}000$	< 1 sec	< 1 sec	1 sec	18 min	very long	very long	very long
$n = 10{,}000$	< 1 sec	< 1 sec	2 min	12 days	very long	very long	very long
$n = 100{,}000$	< 1 sec	2 sec	3 hours	32 years	very long	very long	very long
$n = 1{,}000{,}000$	1 sec	20 sec	12 days	31,710 years	very long	very long	very long

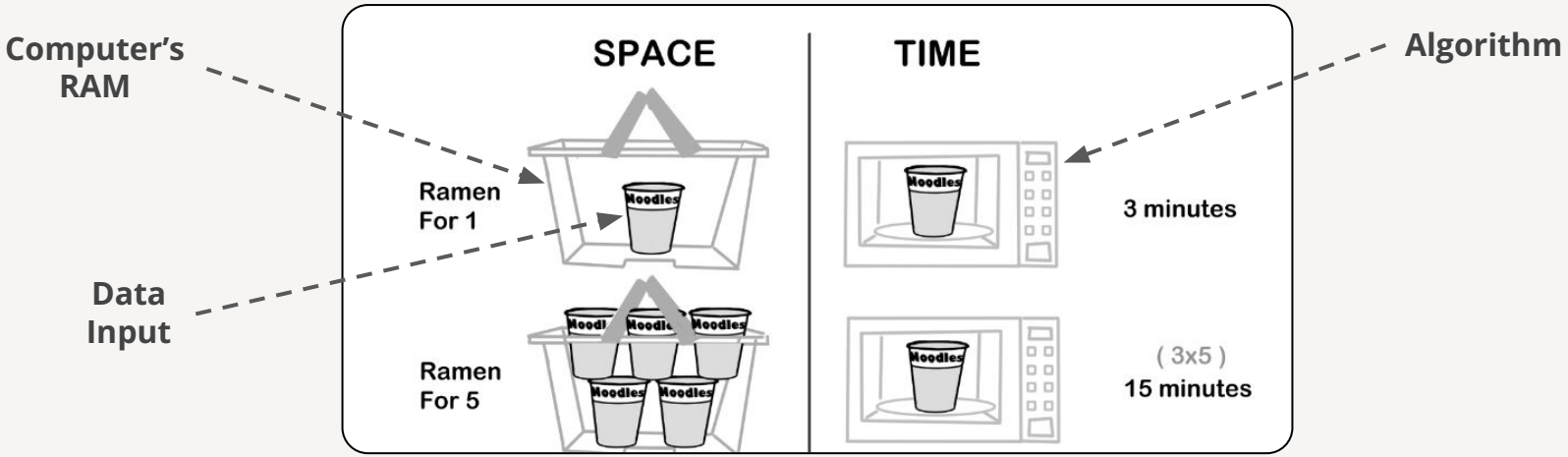
What is Space Complexity?

Space Complexity

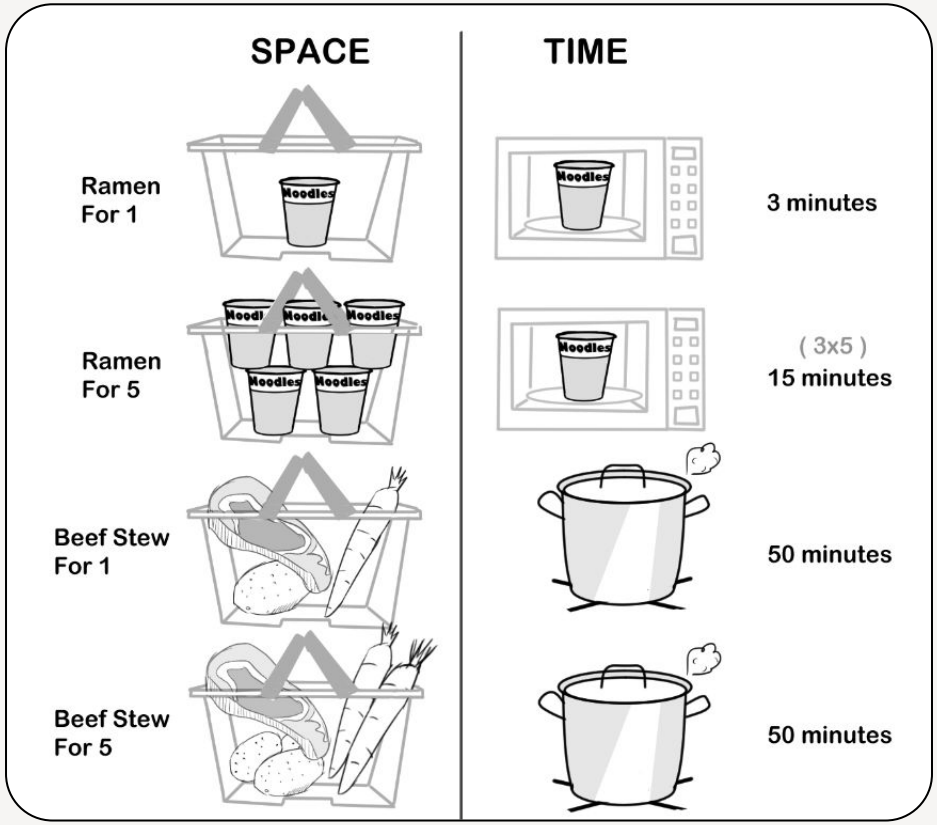
Is usually **referred** as the **amount** of **memory consumed** by the **algorithm**. It is composed of two different spaces; Auxiliary space and Input space.

Space Complexity

Our challenge is that we need to find an optimal way to cook dinner for today



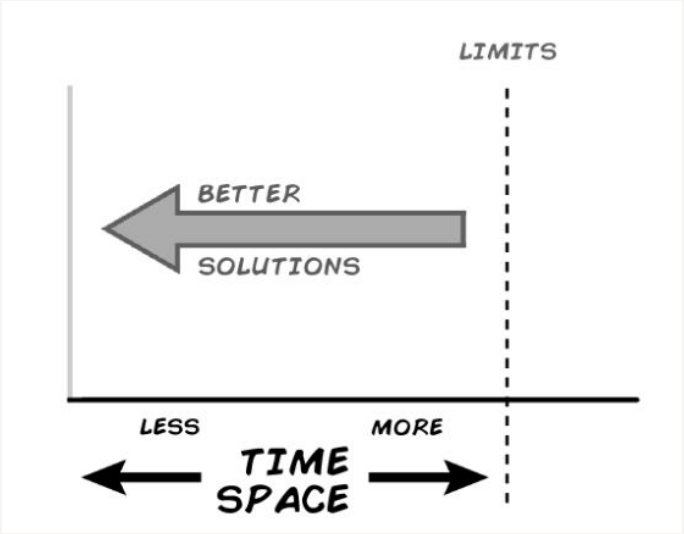
Space Complexity



Space Complexity

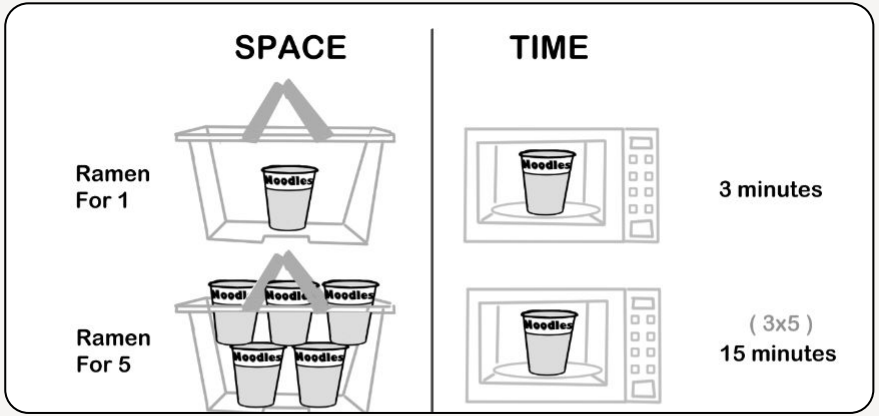
We want to **know how long** will our **code** take to **run** and **how much space** will the **solution use**

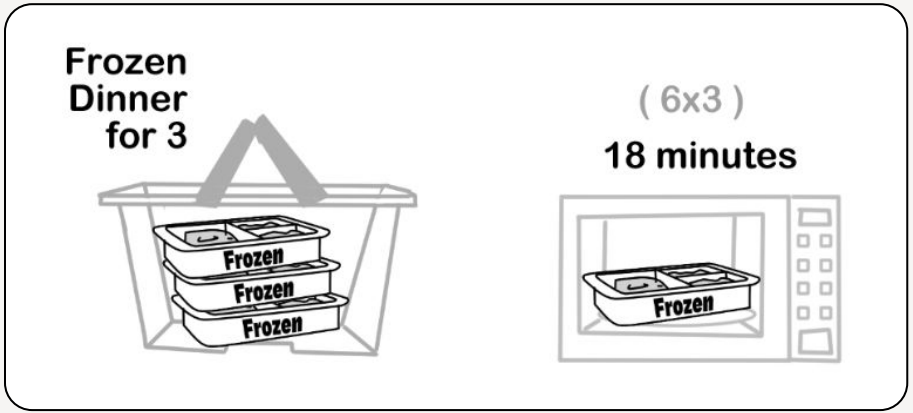
Time (and Space) is Money 🤑



Space Complexity

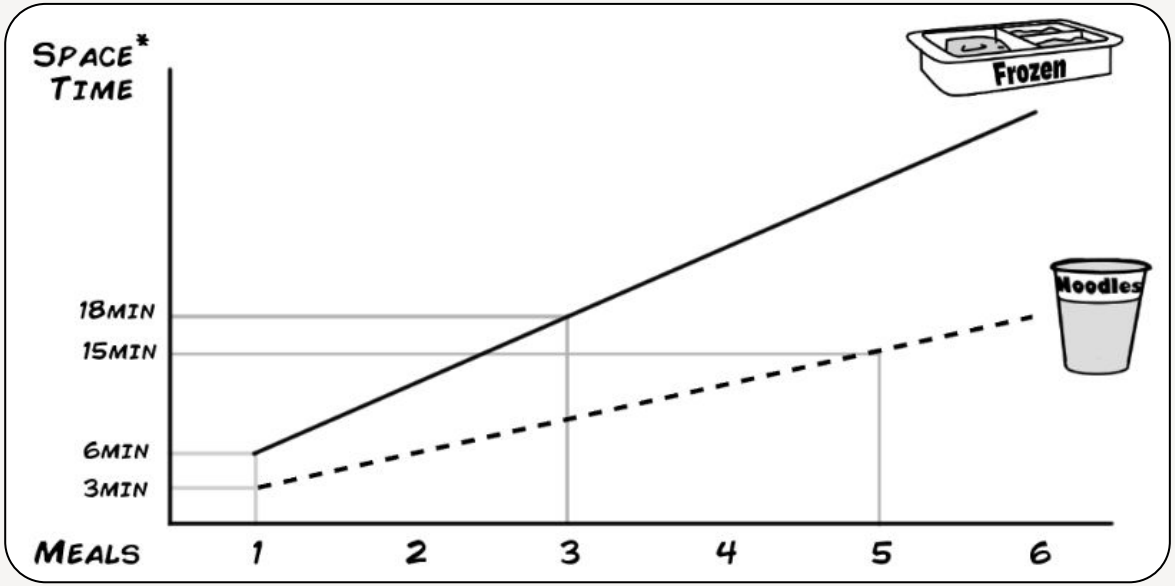
Let's **suppose** that for the **same algorithm** we **change** our **input** or we use a different **data structure** and now **instead** of **3 minutes** we take **6 minutes**.





Space Complexity

Be **mindful** about the **type of input** expected or the **data structure** to **be used** and always **ask yourself** about the **worst case scenario** for the **input** and the expected output.



Infrastructure & Reliability

Infrastructure & Reliability

47% of consumers expect a website to load in no more than two seconds.

Infrastructure & Reliability

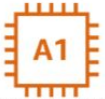
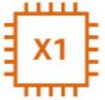
Now a days many cloud providers have different solutions for many use cases and we need to analyze carefully what kind of service we choose.

Infrastructure & Reliability

Although we **have many instances** for **different** types of **complexity** the **more** we push the **vertical growth** the **more** we are going to **pay** for those **resources**.

When to use an EC2 instances?

- **Large** amounts of data.
- **Long** periods of **runtime**.
- **Handle** high types of **complexity**.
- **Impact drastically** your processing **time** with each **type** of **instance.**
- **Scaling** can be **painful** and **takes time**.
- **Manage** the **allocation** of **resources**.

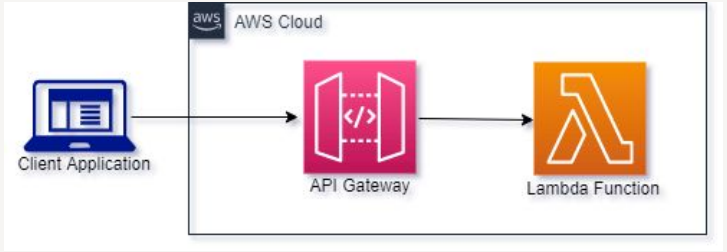
General Purpose	Compute Optimised	Memory Optimised	Accelerated Computing	Storage Optimised
 ARM based core and custom silicon	 Compute - CPU intensive apps and DBs	 RAM - Memory intensive apps and DB's	 Processing optimised- Machine Learning	 High Disk Throughput - Big data clusters
 Tiny - Web servers and small DBs		 Xtreme RAM - For SAP/Spark	 Graphics Intensive - Video and streaming	 IOPS - NoSQL DBs
 Main - App servers and general purpose		 High Compute and High Memory - Gaming	 Field Programmable - Hardware acceleration	 Dense Storage - Data Warehousing

Infrastructure & Reliability

Lambdas are awesome but **have some limitations** on like run up **15 minutes** per **execution**, 10GB of memory and 6 vCPUs

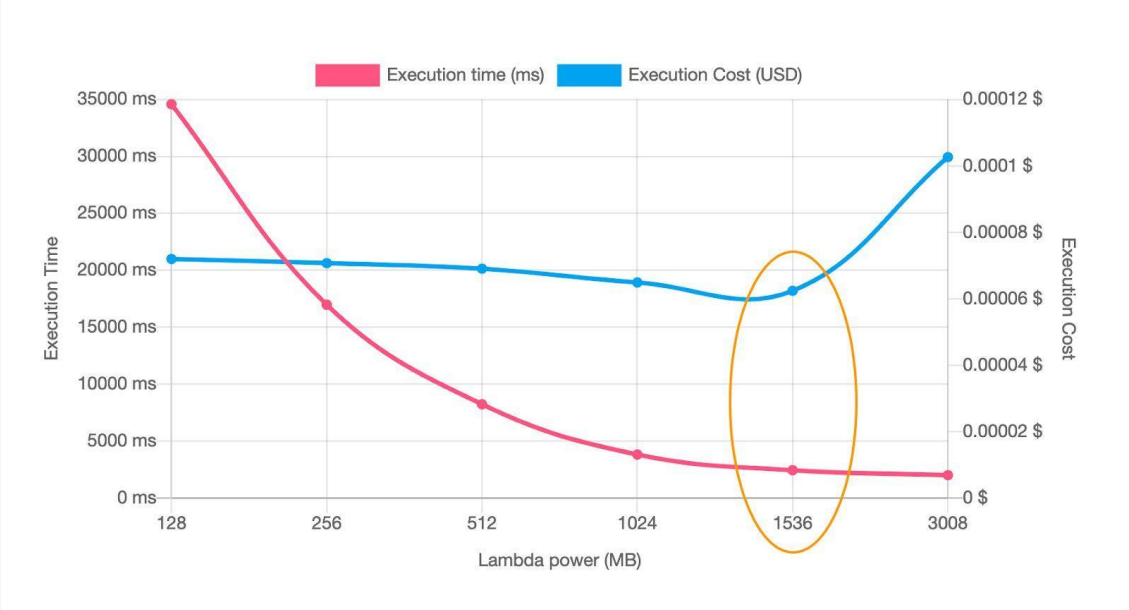
When to use Lambda service?

- **Manage decent** amounts of data but **limited**.
- **Handle short** periods of **runtime**.
- **Handle low** to **medium complexity** BUT be aware of resources and budget.
- **Scales quickly**
- We **don't** need to **manage** the allocation of **resources**
- **API Gateway** timeout is **29 seconds** but it has some perks
- **Include dependencies** can increase **storage limitations** of 50MB zipped or 250MB unzipped



Infrastructure & Reliability

With Lambas is important to find a sweet spot were execution time and memory provide us an efficient solution for our algorithms and provide a great relationship between price and computational power.



Infrastructure & Reliability

Container based technology, Fargate is great option for ease of use as it handles infrastructure management rather than ECS or EKS that delivers more control over the infrastructure.

When to use Fargate service?

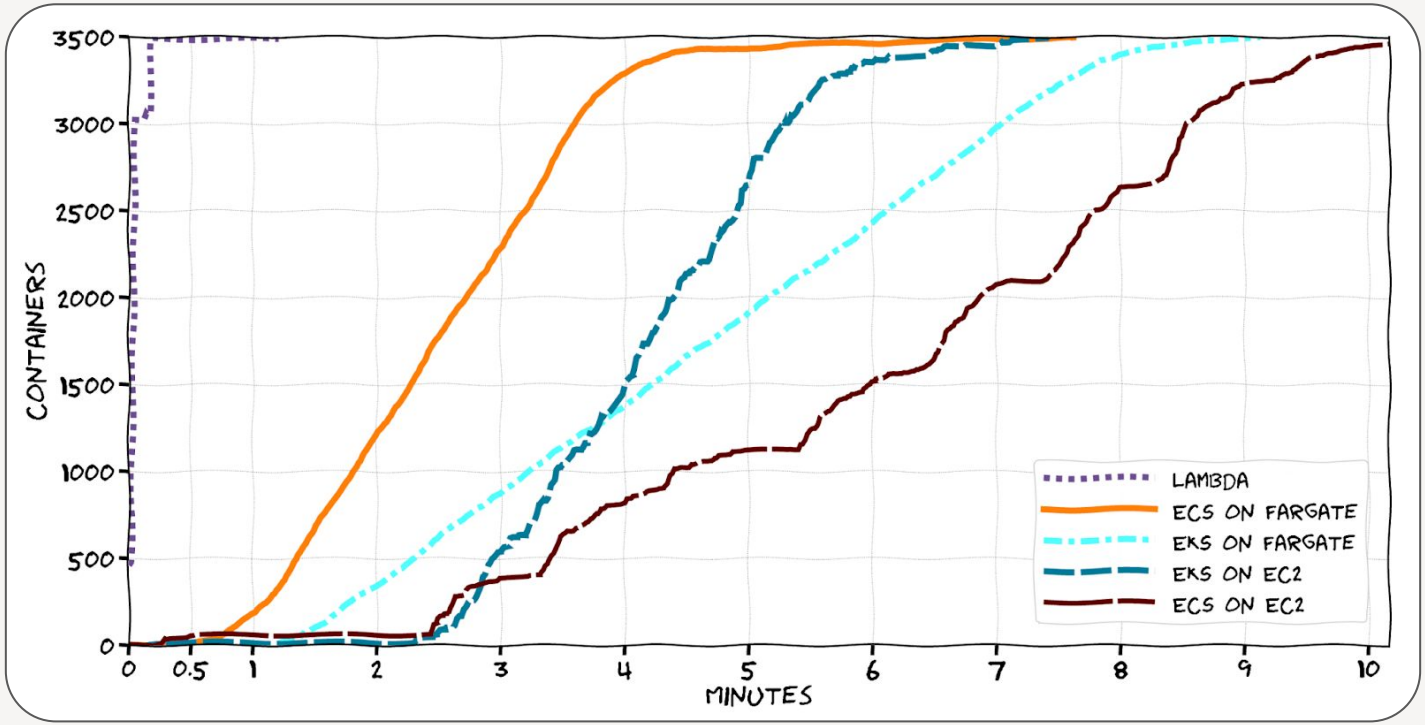
- **Large** amounts of data
- L**ong** periods of **runtime**.
- **Handle** from **medium** to **high** types of **complexity**.
- **Scales quickly**.
- Resources **allocation self managed** in ECS and EKS.
- Avoid **constraints** of **timeouts** but we need to **cover health checks**
- **Containerized** solutions can be **deployed** to **different** technologies

AWS Fargate

Amazon ECS

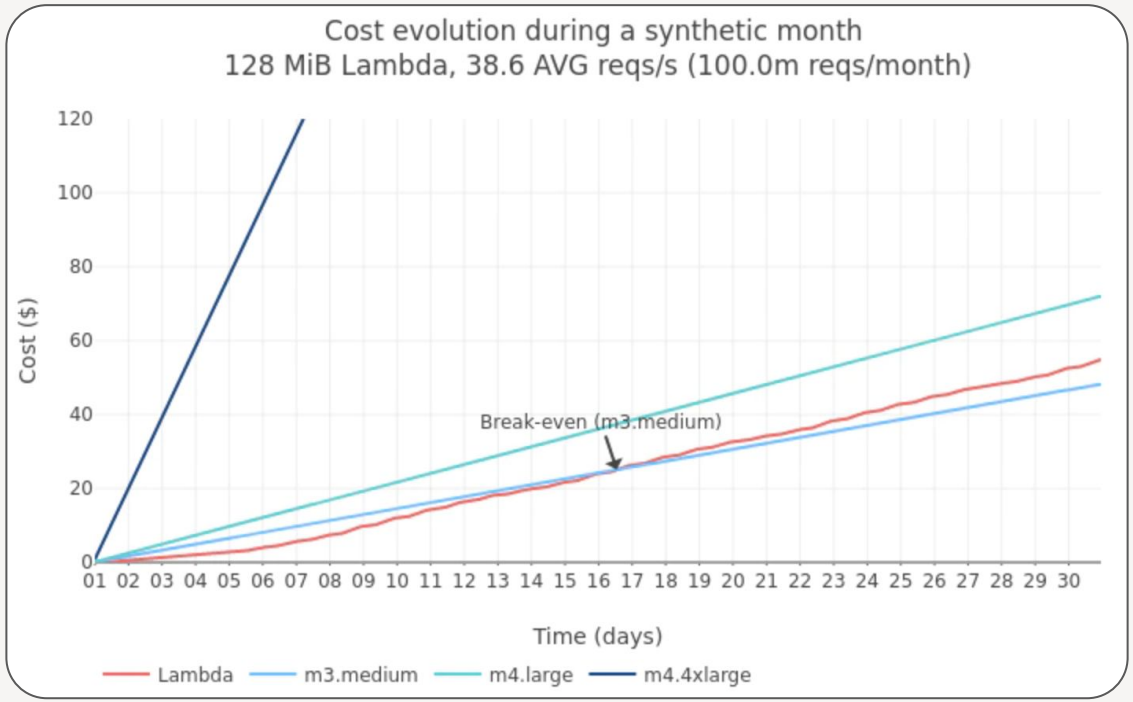
Amazon EKS

Infrastructure & Reliability - Scalability



Scaling containers on AWS in 2022

Infrastructure & Reliability - Pricing



Infrastructure & Reliability - Pricing

	AWS Lambda	AWS Fargate	AWS EC2	Fargate Spot	EC2 Spot
Unit Pricing	$0.0000166667 per GB-second x86	$0.04048 per vCPU-hour + $0.004445 per GB-hour	$0.0416 per hour (t3. medium)	$0.01276392 per vCPU-hour + 0.00140157 per GB-hour	$0.0181 per vCPU (based on estimated 70% spot saving)
Calculation	$0.0000166667 x 4 GB x 60 seconds x 60 minutes x 2000 containers	($0.04048 * 2 vCPUs + $0.004445 * 4 GB) x 2000 tasks	$0.0416 x 2000 instances	($0.01276392 * 2 vCPUs + $0.00140157 * 4 GB) x 2000 tasks	$0.0181 x 2000 instances
Price per Hour for 2000 instances (4GB memory / *2 vCPUs)	$480.00	$197.48	$83.2	$62.26	$36.2

Summary

We need to take in **consideration time** and **space complexity** of our **algorithms** to **determine** the **size** of our **input** data and the **time** it takes to **process** it

- **Understanding** our **application** input and **propose efficient** data **structures** to handle it can be a game changer.
- **Know** in **advance** our time **complexity** can give us **hints** about the **infrastructure** we will need to **implement** in the **long** term
- **Calculate** our **space complexity** can help us to **choose** the **best data storage** for our application.
- Have **everything documented** can save our **projects** time **integrating technology** and **money** to the **business**

Time Space Complexity Implementation Cheat Sheet

Time/Space	**Low**	**Medium**	**High**
Low	Lambda	Lambda Fargate	Fargate ECS
Medium	Lambda Fargate	EC2 Fargate ECS	EC2 ECS
High	Lambda EC2 ECS Fargate	EC2 ECS	EC2 EKS

¡Gracias!

¿Preguntas?

@edm0cha