|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Cloud t2.micro  Instance No Credits | Cloud t2.micro  Instance with Credits | Your Physical  Machine |
| CPU Model and  Speed (GHz) | Intel(R) Xeon(R) CPU E5-2686 v4 a 2.30GHz | Intel(R) Xeon(R) CPU E5-2686 v4 a 2.30GHz | Apple M2 3.49GHZ |
| Total time to run  cpu\_test.py  (seconds) | 0.804732387 | 0.046471572 | 0.010208 |
| Total time to run  cpu\_test.py  (microseconds) | 804732.387 | 46471.572 | 10208 |

A graph with blue line and green line

AI-generated content may be incorrect.

**1. Understanding Cloud CPU Management**

**1.1 What is a CPU credit in AWS?**

A **CPU credit** is a unit used by AWS **burstable instance types (like t2.micro)** to measure how much CPU time you can use above the baseline performance.

* Each credit equals **one minute of 100% CPU usage on one vCPU**.
* When your instance is idle or lightly loaded, it **earns credits**.
* When it needs more CPU (burst), it **spends credits** to temporarily increase performance.

**1.2 How does it benefit the Cloud user?**

It allows users to:

* **Save cost** — you pay for a cheap instance that runs fast only when needed.
* **Burst performance temporarily** — good for workloads that are idle most of the time but need short bursts of CPU (e.g., small websites, microservices).
* **Smooth user experience** — it feels fast even on low-cost instances.

**1.3 How does it benefit the Cloud provider?**

It helps AWS:

* **Optimize resource allocation** — AWS can oversubscribe physical CPUs because not all burstable instances run at 100% simultaneously.
* **Increase utilization efficiency** — idle periods from many users let AWS host more instances per physical host.
* **Offer low-cost options** — attracting users who don’t need constant high CPU performance.

2. Comparing Scenario 1 vs Scenario 2

Scenario 2 (EC2\_t2micro\_fresh) has **better CPU performance** because it still has CPU credits to burst above the baseline, while Scenario 1 is throttled once credits are exhausted.

3. Comparing with Your Local Machine

Your **local notebook is faster** than both t2.micro instances.  
Reason: your physical machine likely has more CPU cores, higher base frequency, and no credit throttling like AWS t2.micro (which has only 1 vCPU and limited burst performance).

4. CPU Utilization Explanation

Only the **fresh t2.micro** and your **local machine** reach near 100% utilization.  
The **depleted t2.micro** is throttled by AWS to maintain fairness and limit usage after credit exhaustion.