Cloud computing is all about flexibility and efficiency, yet many engineering teams still get tripped up by misconceptions around shared CPU instances. Whether you're running microservices, worker queues, or internal tooling on AWS, Azure, or Google Cloud, these misunderstandings can lead you into expensive overprovisioning or, worse, unstable performance.

In this post, I'll share what I've learned over 12 years managing cloud infrastructure and running cost reviews — debunking common myths around "low average CPU usage," "vCPU guarantees," and "bursting" behavior, especially on shared CPU instances. Along the way, I'll reference how tools like **AWS Compute Optimizer** and **Azure Advisor** can help—if you use them the right way.

Why Shared CPU Instances Are Challenging to Understand

Before diving into misconceptions, it's important to clarify what *shared CPU* means and why it isn't a uniform standard across cloud providers.

Shared CPU Definitions Differ by Provider

AWS, Azure, and Google Cloud each have nuanced definitions of what a "shared CPU" or "burstable" instance type means. For example:

- **AWS** defines T-series instances (like t3, t4g) as *burstable performance instances* with CPU credits that accumulate during idle periods and can be spent during CPU spikes.
- **Azure** offers B-series VMs, also burstable, but their CPU credit accrual and spend logic differs from AWS in credit duration and throttling behavior.
- **Google Cloud** provides shared-core instances that allocate fractional vCPU time slices, with distinct scheduling algorithms.

This means you can't assume the same performance characteristics or billing implications across providers. What counts as "shared CPU" on AWS may not offer identical CPU bursting or throttling behaviors compared to an Azure B-series instance.

Myth 1: The Low Average CPU Usage Myth

Perhaps the most common pitfall is interpreting **average CPU utilization** as the full picture of an instance's workload. It's tempting to look at 10% average CPU over 24 hours and conclude your workload is lightweight enough to run on a small burstable instance...

...but this ignores the critical fact that cloud performance constraints—and costs—are tied to CPU *peaks*, not averages.

Why Averages Don't Cut It

Average CPU usage is a smoothing operator: it hides spikes and valleys in utilization. A worker queue that processes bursts of job arrivals every few minutes might show 15% average CPU but actually hits 90% CPU for short intervals. Running on an instance too small to handle those spikes can cause backlogs, timeouts, and retries that silently reduce SLA compliance.

The Right Way: Measure with Percentiles and Spike Duration

Instead of relying on averages, focus on percentile metrics like P95 and P99 CPU usage over meaningful observation windows:

- **P95 and P99:** Show CPU utilization levels that 95% or 99% of the time your workload will stay under—key to designing capacity for peak demand rather than average.
- **Spike duration:** How long do those CPU usage peaks last? One-second spike versus sustained five-minute strain have drastically different infrastructure implications.

If you're using AWS CloudWatch or Azure Monitor, set custom metrics to capture percentile CPUs and use those to feed tools like AWS Compute Optimizer or Azure Advisor for tailored sizing suggestions.

Myth 2: The vCPU Myth — More vCPUs Equals Better Performance Guarantees

Another widespread misconception is treating virtual CPU (vCPU) counts as strict performance guarantees. For instance, choosing a 4 vCPU burstable instance and assuming you'll consistently get 4 full CPU cores' worth of compute during work spikes.

This is fundamentally flawed because a vCPU in burstable/shared CPU instances doesn't mean an entire physical core reserved for your workload. Many providers oversubscribe physical cores, meaning your "virtual core" shares cycles with other tenants' workloads.

Why vCPU Counts Are Not Performance Guarantees

- **Shared Scheduling:** On shared CPU instances, vCPUs are scheduled on physical cores along with other instances, which can cause performance variability.
- **CPU Credit Models:** Burstable instances rely on CPU credits to allocate CPU time slices. Exhausted credits can throttle your effective CPU well below the vCPU's nominal capability.
- **CPU Steal Time:** Cloud hypervisors can "steal" CPU time when resources are needed elsewhere, impacting your vCPU performance unpredictably.

This means engineering teams must measure the *effective* CPU availability and throughput during peak loads rather than blindly selecting instance types by vCPU count alone.

How to Navigate vCPU Ambiguity

Ask these questions before instance type decisions:

1. What does the P95 and P99 CPU utilization look like over my workload's busiest windows?
2. How long do CPU bursts sustain during those peaks?
3. Do CPU credits accumulate and deplete in a pattern that aligns with my burst workloads?

This mindset changes the conversation from "How many vCPUs do I need?" to "What observed CPU capacity do my workloads actually consume at peak, documented and verified?"

Myth 3: The Bursting Myth — Burstable Instances Will Always Cover Your Spikes

Engineers love burstable instances because they promise cost savings by running small with the ability to burst when needed. However, the "bursting" story glosses over important operational realities.

Bursts Are Neither Constant Nor Unlimited

Burstable instances accumulate CPU credits when idle, then spend them to handle bursts. But:

- **Once CPU credits run out, your instance is throttled** to baseline CPU performance, which can be inadequate under load.
- **Bursts are designed to be short-lived.** Sustained high CPU usage will quickly drain credits, causing throttling that can bottleneck your workload.
- **Credit accrual rates & baselines differ** widely even between AWS T3 and T4g or Azure B-series VMs, so you can't assume equivalence.

What Bursting Myths Mean in Practice

Teams often deploy always-on, small burstable services — such as monitoring agents, small internal tools, or staging environments — assuming they are a cloud "free lunch." Yet these "low-CPU average" services mask cloud waste because:

- They accumulate CPU credits mostly unused, representing stranded capacity.
- Sporadic bursts might cause hidden throttling events, degrading user experience or queue latency.
- Compute Optimizer or Azure Advisor recommendations based on averages can misleadingly suggest further downsizing.

Don’t accept average CPU as the only input. Instead, verify if your burst Windows and credit models align with your workload’s real burst patterns.

How to Use AWS Compute Optimizer and Azure Advisor Effectively

Tools like AWS Compute Optimizer and Azure Advisor are valuable—but only when armed with the right data and observation mindset.

Feed Percentiles and Peak Metrics, Not Averages

Both tools pull instance metrics to recommend resizing or switching instance families. If those metrics represent only average CPU usage, recommendations risk excessive downsizing or ignoring burst requirements.

- **Customize CloudWatch Metrics (AWS):** Collect CPUUtilization P95/P99 percentiles over a representative window and export to Compute Optimizer.
- **Configure Azure Monitor:** Use Azure Metrics Advisor or advanced analytics to track CPU percentiles and burst duration for feeding Azure Advisor.

Set Proper Observation Windows

CPU usage varies hourly, daily, and weekly. Capture at least 7-14 days of workload performance including peak demand periods. A few hours of low load will skew averages dangerously low.

Include Storage and Egress Costs in Cost Reviews

CPU utilization is only part of the cloud cost story. Don’t ignore storage and data egress when sizing instances for internal tools or worker nodes—because these can dwarf CPU costs or introduce hidden throttling from network saturation.

Summary Table: Myth vs Reality

Myth Common Misunderstanding Reality Practical Advice Low Average CPU Myth Average CPU usage indicates safe instance downsizing. Average hides peak CPU spikes critical to performance. Use P95/P99 CPU and spike duration to guide sizing. vCPU Myth More vCPUs equal guaranteed compute performance. vCPUs on shared instances are time-sliced and oversubscribed. Measure effective CPU availability during peaks. Bursting Myth Burstable instances always cover workload bursts. CPU credits are limited; burst duration is short. Understand credit models; monitor throttling and credit balance.

Final Thoughts: Rollout Safely with Data-Driven Decisions

When migrating or optimizing workloads on shared CPU instances, make sure to:

1. **Measure peak CPU usage** with percentile metrics and appropriate observation windows.
2. **Don’t treat vCPU counts as hard performance guarantees.** Understand provider-specific oversubscription and credit models.

3. **Use AWS Compute Optimizer and Azure Advisor** as guides—not gospel—and feed them high-fidelity metrics.
4. **Document rollback criteria** before resizing or resizing pilots, so you can revert if bursts exceed capacity.
5. **Factor in storage and egress costs** to avoid getting cost surprises beyond CPU.

Ultimately, using shared CPU instances effectively requires thinking beyond simple averages and counting cores. Pay attention to the *peaks, the percentiles, and the burst duration*. This engineering discipline helps you reduce cloud waste without sacrificing reliability.

If in doubt, run a pilot for your workload, monitor P95 and P99 CPU in real-time, and only then adjust instance computingforgeeks.com sizes. That's what's worked for me across multiple large-scale cost reviews and migrations.