silk

How to Accelerate Cloud Performance

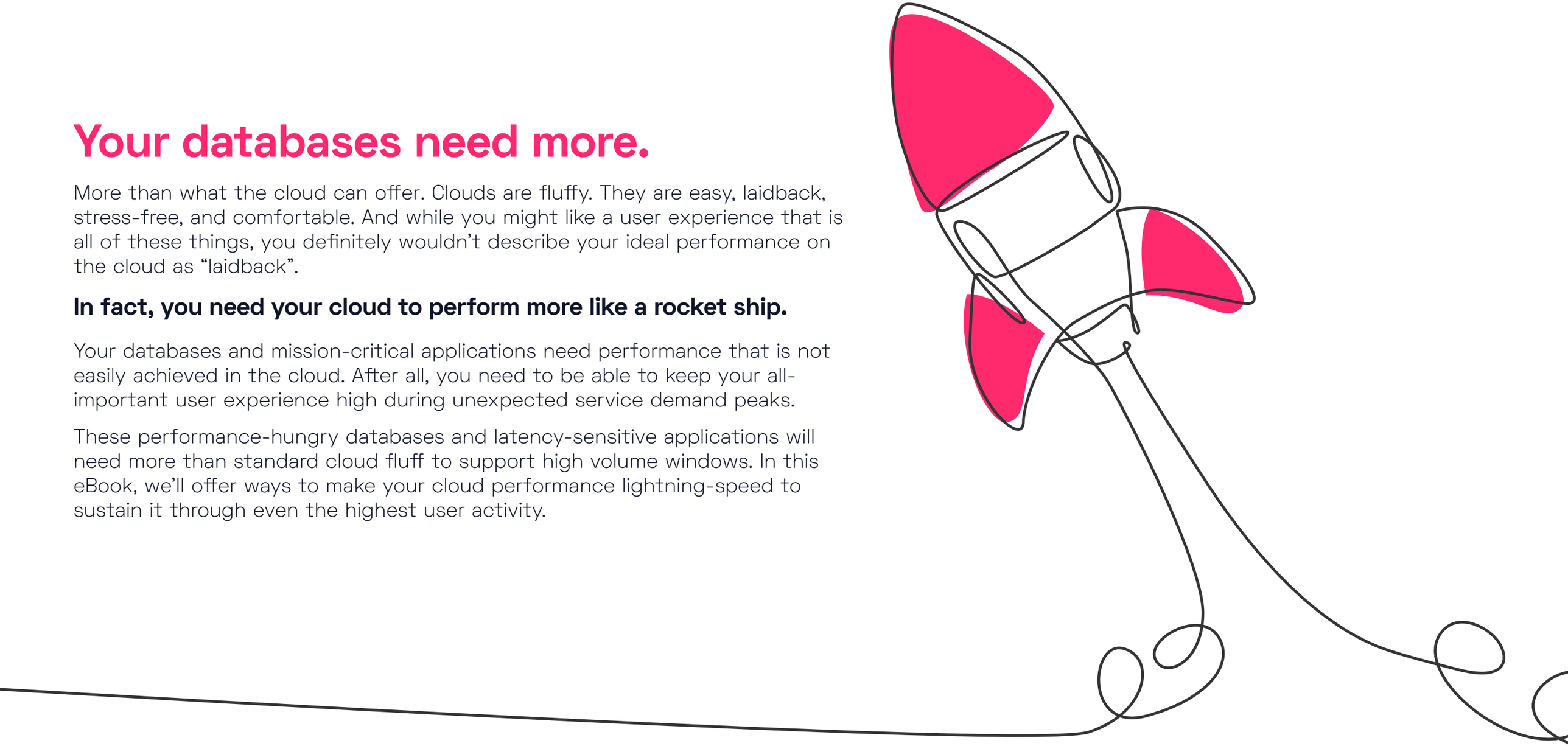
Your databases need more.

More than what the cloud can offer. Clouds are fluffy. They are easy, laidback, stress-free, and comfortable. And while you might like a user experience that is all of these things, you definitely wouldn't describe your ideal performance on the cloud as "laidback".

In fact, you need your cloud to perform more like a rocket ship.

Your databases and mission-critical applications need performance that is not easily achieved in the cloud. After all, you need to be able to keep your all-important user experience high during unexpected service demand peaks.

These performance-hungry databases and latency-sensitive applications will need more than standard cloud fluff to support high volume windows. In this eBook, we'll offer ways to make your cloud performance lightning-speed to sustain it through even the highest user activity.



1 Select the Right Data Platform

Cloud vendors offer many different types of storage. The highest performing storage media is flash, namely: SSD PD from Google Cloud Platform, Elastic Block Store (EBS) from Amazon Web Services (io1/io2), and Ultra SSD from Microsoft Azure. This flash media will deliver from 64k up to 100k IOPS and a maximum throughput of up to 1-1.2 GB/s. The latency (the time delay in how long the storage receives an IO request until it fulfills that request) is supposed to be between 1-9ms on average as a generic SLA.

These numbers are perfectly adequate for a great many applications, but often times not enough for transactional or analytic database applications where time to insight and time to action are the key metrics for success. How fast can we ingest, analyze, and deliver a valuable action back to the customer? For many applications (more and more every day) the answer is "As close to real-time as possible". This would enable richer and more satisfactory client engagements right at the edge where people live and work today.

Standard cloud infrastructure can really struggle to deliver this kind of great user experience consistently without introducing major imbalances in cost and effort to outcomes.

To break these limitations, you need to break away from the "shared nothing" cloud architecture to adopt one that is built from the ground up to aggregate performance without any limits.

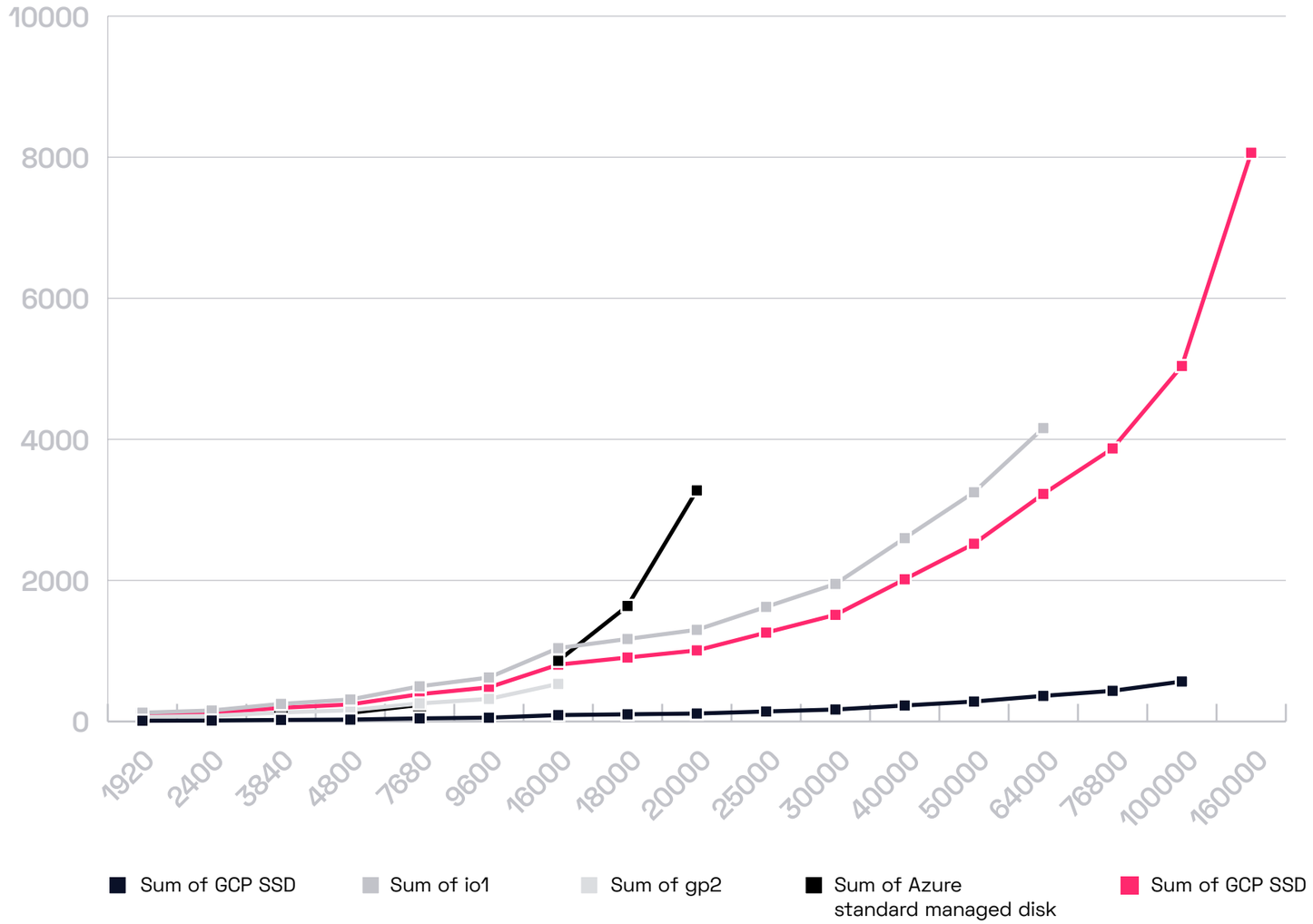
FAST FACTS: Flash media on the public cloud delivers 64k-100k IOPS, maximum throughput of 1-1.2 GB/s, and latency of 1-9ms

2 Learn the I/O-Capacity Tango

IOPS are a measure of how many operations the platform can execute in a single second. This measurement is a bit more complicated than a simple single number since a dozen factors will move that number up or down significantly. Simplifying it: in the cloud, there are two ways to get the amount of IOPS your applications need. First, some of the cloud vendors will provide IOPS that vary depending on the amount of capacity or volume size you provision and pay for. So, if you need more performance? Easy—just continually add more media capacity until you finally get enough IOPS… or until you hit the maximum allowed. If that's not enough, you're kind of stuck until you re-architect your solution :). If you can get enough IOPS by adding more capacity, just remember that you'll be paying for all that extra capacity—whether you use it or not.

Some cloud providers have the option to pay extra for provisioned IOPS. You can allocate the capacity you need but pay an additional cost for a fixed amount of provisioned IOPS. Again—whether you are using them or not. You're paying for the maximum option each time.

Both of these options can be quite expensive and often result in wasted or stranded resources that you pay for every minute, but rarely (if ever) use.



3

Not All IOPS Are Equal

Confusingly, IOPS calculations are not straightforward with the formula to derive a valid number having a lot of variables in it. Cloud providers use a fixed I/O size and a baseline number to show a possible maximum number of IOPS that can be achieved, but the real max number changes depending on your actual workload. For example, EBS provisioned IOPS are calculated for IOPS with a block size of 16 KiB. But if your application uses a block size of 32 KiB, you will only get 50% of the provisioned IOPS you allocated.

As block sizes increase, IOPS decrease. Since all databases and applications use many different block sizes for various actions, calculating the needed amount of provisioned IOPS can be a real challenge.

4 Performance Consistency

Performance guarantees among the major providers are not a guarantee at all. Latency on a volume can range from single digit milliseconds into the hundreds of milliseconds. Your variable workloads (and other workloads leveraging shared components of the infrastructure, aka "noisy neighbors") cause massive latency fluctuations, leading to inconsistencies in application response times, which makes end users quite frustrated.

To deal with performance fluctuations, you may need to overprovision (and pay for) resources to get enough top-line performance to make sure you have enough headroom exactly when you need it. Plan for the worst, and then add some more. This is called "architecting for the peak" and it can be quite expensive, not to mention inefficient.

The "Benefits" of Overprovisioning

Current public cloud architectures have resources that are designed to be fairly tightly coupled to each other, impacting the ability to scale in a granular fashion. For example, the type and size of the compute engine in CPU and DRAM memory will also determine the amount of network bandwidth or storage performance you can provision as well. It's a three-legged stool with not a lot of room for movement in the amount or ratio of resources to each other. As a result, a client will need to provision far more of a certain resource than the application actually requires to get enough of another resource that the application needs. This overprovisioning is costly, and quite common, especially in applications that are resource "hungry" for performance or the number of vCPU cores. The downside is that the client is paying for 100% of every allocated resource, even if they are only using 10% of them, as resources are paid for when they are turned on, whether they are actively being used or not.

This is a very common source of frustration among cloud customers, as cloud native resources do not let clients pick and choose the specific resources they need and use them as flexibly as they would like. It's more of a bundled or pre-packaged approach, with some flexibility within small ranges, but when requirements begin to scale, the imbalance between resource allocations becomes significant, and costs become unreasonable. While most clients are fine accepting some overprovisioning waste and cost burden, it is a hard-limiting factor when significant expansion, scaling, or the need to bend cost curves factor in — almost always the case for real Tier 1 production applications (not just test/dev environments). Clients want to benefit from an economy of scale, not be penalized for it.

By choosing a platform that decouples performance from capacity, you can independently scale one without the need to scale the other – eliminating the need to overprovision.

5 Old School Solutions

One legacy way to boost disk performance is to configure multiple SSDs with software based RAID0 (multi-disk striping). You'll get additional aggregated performance in terms of IOPS and bandwidth, with the extra cost of each additional volume you connect. This comes in handy when your volumes reach performance limitations.

This solution also brings significant risk to data resiliency and availability, as losing any disk in the stripe results in total data loss of the entire volume. Software RAID also creates local CPU overhead and configuration efforts.

Silk offers any performance to any volume regardless of its size, while securing your data under a fully automatic RAID-6 with patent-protected technology that offers exceptional utilization of 87.5%.

Relational Databases in the Cloud

Users who have previous experience with relational databases on-premises find that the maxed-out performance on the cloud leaves much to be desired, while the capacity-performance pricing model leaves you with capacity you don't need.

One born-in-the-cloud analytics company was experiencing this exact issue with their two MongoDB environments in GCP. Given the complexity of its databases, it needed to run multiple copies of their entire database 24/7 to achieve a high level of performance.

With the Silk Platform, the company saw 2x greater performance and a 70% cost savings due to the platform's dramatically improved performance (versus native cloud alone) and Tier 1 data services that consolidated data replicas.

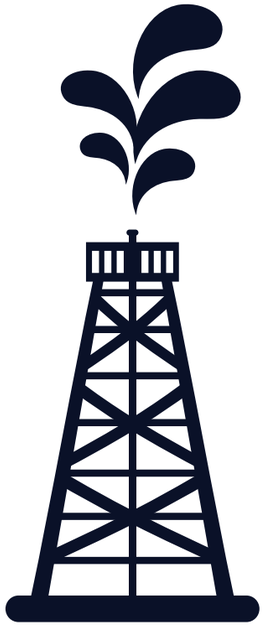
6 Use Snapshots Wisely

Working with snapshots can challenge your performance in multiple ways. For example, with EBS you will need to use FSR (Fast Snapshot Restore) to get full performance during restore—for an additional cost, of course. In some cases, creating snapshots will impact your data performance. Normal snapshot and restore/clone operations can take a long time—30 to 60 minutes or longer depending on how much data you are cloning. Storing snapshots is not free either, with each snapshot costing you more. Cloning a snapshot to a new volume incurs new full costs for that volume.

7

Compute Instance Bingo

Choose the right type (from the myriad choices) of compute instance to optimize your performance. This can be really quite difficult and will dramatically impact data performance. Compute instances have their own highly varied performance limitations, which must be considered. Try using EBS-optimized instances in AWS, adding more vCPUs to your VM in GCP, or the required size/type of your VM in Azure. If you are hitting the caps on IOPS or throughput with your specific compute engine, you'll need to upgrade to a more powerful class.



Soccour Solutions: A Real-World Example

Business Solutions Provider, Soccour Solutions, needed a flexible SaaS solution for its client during the COVID-19 pandemic. The solution needed to offer high-performance in order to meet the client's 3D visualization demands and to run high volumes of data.

With the Silk Platform on AWS, the software ran 40-60% faster than it had run previously on on-premises physical workstations. And compared to native AWS, where they were achieving maximum speeds of 1.6MB/s, Silk offered Soccour Solution's client 400MB/s.

8

A Comprehensive Solution

Are you using all these tricks, paying for all these extras, and still hitting a performance ceiling with your workloads and not getting the required low latencies for your critical applications?

Try the Silk Platform. It will make your life easier with dynamic ultra-high performance at a lower cost. Silk solves all of these cloud limitations with an 8th generation enterprise class software data platform.

How? Silk dynamically virtualizes multiple disks and aggregates the performance for the application layer while applying patented unique algorithms to optimize the utilization of the underlying hardware. With Silk, you can dynamically get any performance you need without overprovisioning capacity — and pay exactly for what you use.

With Silk's dynamic performance and consistent latency, getting your performance is as smooth as silk. At the same time, you'll pay for only what you use without the need to plan and pay ahead for peak workloads.

With Silk, snapshots are zero-footprint and instantaneous with no penalty on performance during snapshot creation or restore — and no additional cost.

"On native AWS, we were seeing a maximum speed of 1.6Mb/s. But with Silk on AWS, we saw 400MB/s. That increase in speed really is incredible!"

- David Duncan,
Principal Solutions Engineer,
Soccour Solutions

Ready to see how the Silk Platform can dramatically improve your performance in the cloud? **Visit www.silk.us.**