SOLUTION BRIEF

Online Gaming Back-End Infrastructure and LightOS™

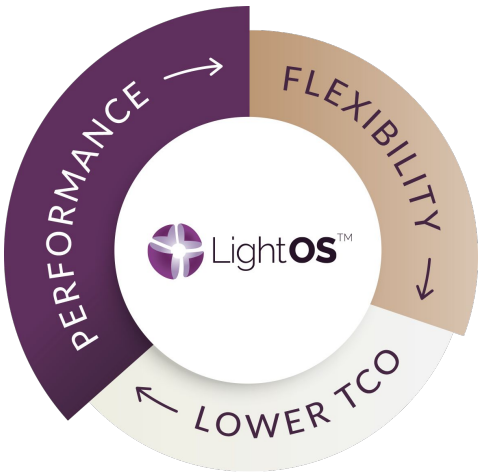
INTRODUCTION

In the world of online gaming there are many enemies. To players, the word "enemy" might conjure up the image of a monster, the uniform of a hostile combatant or the weapon of a favorite foe. There is one enemy that can appear in any game, on any platform, and is one of the most troublesome in existence - lag.

Lag is the Massively Multiplayer Online (MMO) enemy in first person shooters (FPSs), real-time role playing games and real-time strategy games. Lag is a noticeable delay between the action of players and the reaction of the back-end server(s) supporting the game returning to the client during gameplay. Because these types of MMOs strive to emulate live reactions, role playing and action simulation, they are highly sensitive to lag.

In more technical circles, we know lag as latency. Latency negatively affects gameplay and hence latency is the enemy in the back-end infrastructure of the MMO provider as well.

WHY LIGHTBITS?



Low latency storage for cloud native databases, Apache Kafka streaming and log analytics. Simplify high performance, low latency, scalable storage that boosts performance, eliminates lag and lowers cost

We talked to various leading MMO providers in preparation of this document. During the course of our interviews we found common themes around back-end infrastructure and concerns around that infrastructure.

While the MMO provider can't control individual gamer latency due to that gamer's internet connection, firewall, or router, they can control the latency of back-end systems. Any latency introduced in back-end systems is magnified by the effects of additional latency to and from the game-client.

So what does this all mean for companies providing back-end services for MMOs? It means it's critically important for them to provide the lowest latency services they can, in the most efficient way possible.

THE PLAYER'S PERSPECTIVE

When we say latency is the enemy of MMOs, we mean that it's a silent enemy that can frustratingly affect gameplay in a variety of ways.

Among the worst effects are:

- Making gameplay feel unrealistic, causing "jumps" and sudden changes in game state.
- Being an advantage or disadvantage (in games like FPSs) to players, as their in-game location may be unpredictable.

In role-playing games, immersion in alternate realities may be lost due to these effects and lead to dissatisfied customers. In our social media dominated society, there is nothing worse for a gaming provider than for a tweetstorm to erupt with dissatisfied players complaining "I just got kicked out of the game!", with hundreds or thousands of "Me too" replies.

Latency above just 100 milliseconds (ms) (1/10th of a second) can have noticeable effects on gameplay. When the internet connection between the player and the MMO back-end servers is between 20 and 50 milliseconds, that could leave only 50 milliseconds on the back-end to process the player input, make a decision about it, and then send back a response.

> *"In a first-person shooter (FPS) like Call of Duty, completing an action has [a] tiny deadline, and latency of more than 100 millisecond[s] can affect the experience of the gamer."* - Venturebeat, April 2016

In MMOs that allow for forming teams or squads, there is a desire for groups of players to form ever-expanding groups. This results in more input to the back-end that must be processed simultaneously and may impact play if it results in more latency. This is mitigated by the MMO providers by limiting team sizes but this decision aimed at improving gameplay can frustrate players that want increased team sizes.

Teams make MMOs fun but can be hindered by traditional back-end storage

MMO PROVIDER BACK-END INFRASTRUCTURE

When breaking down the components of the back-end infrastructure for MMOs we found that it's not so different from the components of enterprise applications, financial services or your favorite social media website. It's custom code, accessing various databases, passing messages and providing responses based on inputs. The key differences are the amount of parallelism (simultaneous, multi-player) and the criticality of fast response time (low latency).

DEPLOYMENT ENVIRONMENT AND APPLICATIONS

COMPUTE AND DEPLOYMENT ENVIRONMENT

For deployment and provisioning environments, some MMO providers use bare metal, some use virtual machines (via Openstack) and others use Kubernetes container environments. It's not unusual for these providers to have a mix of these deployment methodologies.

All MMO providers used Linux in their back-end environments with some having pockets of Microsoft Windows Server. Physical deployments incorporate protection domains such that (for example) an entire rack or row can fail and the back-end services will still stay up, as uptime is critical.

PROVISIONING, MANAGEMENT AND MONITORING

In order to support their internal development groups, most support groups have moved to cloud-style self-service portals for requesting and allocating resources. To support such functionality or to simply be integrated into any provisioning automation, a robust and flexible API for storage is mandatory. Additionally, operations groups have their own tools for monitoring and management and therefore there is a need to integrate with those systems rather than to force the usage of a proprietary GUI or other static or inflexible tools.

APPLICATIONS

On the back-end applications side, providers reported using common databases such as MySQL, MongoDB, Apache Cassandra, and Microsoft SQL Server. It was common for these databases to be multi-terabyte. A player action or client request can result in multiple database queries measured in single digit milliseconds. The underlying database storage must therefore be as responsive as possible as it is at the bottom of the complete stack. Any delay in storage access will propagate up the stack and all the way back to the player.

Intimately linked to databases, real-time analytics and telemetry are streaming data infrastructures. Some use custom message/streaming protocols, and others use Apache Kafka. This makes the data streaming environment highly dependent on low latency storage as well.

COMMON GROUND

What MMO providers all had in common was the decision for any back-end storage to be based on flash - no "spinning rust" when low latency and predictability is paramount. One provider went so far as to tell us that their goal was for all back-end storage to respond to small block queries in 1 millisecond or less, all of the time.

FLEXIBILITY

Gaming back-ends need to be flexible and dynamic. To lower latency, they are often deployed in edge data centers to lower latency to/from gaming clients. This flexibility requires storage solutions to be software-based, potentially running on various configurations of hardware based on location, environmental restrictions or budget. Additionally, storage solutions for gaming must be scalable to deal with sudden needs or planned growth of a platform due to popularity surges, or simply a new release.

Some companies buy standard servers in advance so they can "make mistakes" - that is, they may not purchase the exact configurations that they end up needing with the notion that software defined solutions can overcome many shortcomings in procurement. Lastly, advanced storage features such as thin provisioning are nice to have, but they will trade thin provisioning for thick if the feature significantly impacts latency performance.

BURST NEEDS

Gaming back-end infrastructure support groups aren't just responsible for the infrastructure supporting game play, but also the internal groups doing development, analytics and release management. At a moment's notice, a development group can say "I need 50 compute servers and 50TB of storage [all flash] that needs to be fast". This need can be for special development, or for an alpha or beta deployment of a game requiring a complete, independent back-end. If beta, it might require multi-geography deployment.

While this sounds like a perfect candidate to deploy in the cloud, it has proven to be expensive and falls short of expectations. Most storage options in the cloud have difficulty meeting the required performance needs or are very expensive to do so. Secondarily, experience has shown that while a group might request 50TB, they won't use that storage for some time - possibly never. As cloud storage is charged on a thick provisioned basis, this becomes a highly expensive endeavor. Some companies feel they can save money by provisioning on-premise for these unique needs.

"Up to 58% of organisations said that moving to the cloud has been more expensive than initially thought."
- Capita Consulting, February 2020

FUTUREPROOF

Online gaming back-end infrastructure must be future proof. It may run for a long time and/or be re-used for other games or development purposes. Storage is a major deciding factor here. The notion of deploying local flash (Direct Attached Storage, DAS) in each application server may be attractive from a performance and convenience standpoint but requires a decision: what size drive?

Too large, and the capacity may never be fully utilized. What's "just right" today may be too small tomorrow and hinder reuse or require more servers than is necessary to have enough total storage. Servicing systems with DAS becomes an increasing operating expense over time.

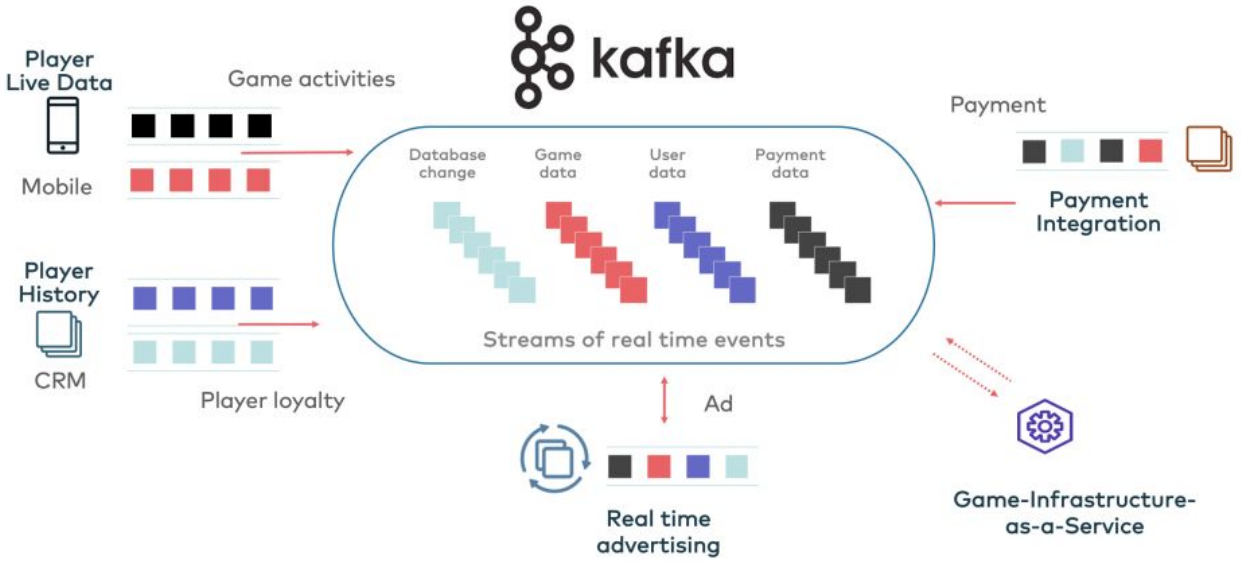
The argument against disaggregation until now has been performance (especially latency) and cost. No one wants to go back and retrofit old servers with larger drives. The obvious answer is to disaggregate compute and storage for independent scaling and to allow for longer use of application server assets.

USER DATA LOGGING AND TELEMETRY

Logging user data (back-end) on gameplay is highly valuable to analyze gameplay and in turn, that analysis can be used to make the game(s) better. Turning on logging at a granularity that is useful requires high IOPs at low latency for large clusters, yet this logging cannot impact gameplay.

Sometimes, a decision is made to reduce or turn off logging to preserve gameplay. This means that data scientists and game developers lose out on precious data, thus hindering improvements. Analysis of the logs and telemetry results in new features or other game improvements that would otherwise be lost, or based on guesswork.

Telemetry (often incorporating Apache Kafka) can be used in monetisation networks for real-time advertisement and "in-app" purchases, real-time financial fraud detection or cheating detection. For any of these activities, latency matters and the message streaming services and databases used for these activities are highly dependent on their underlying storage.



Apache Kafka in the Gaming Industry – @KaiWaehner - www.kai-waehner.de

KAI WAEHNER

LIGHTOS MEETS AND EXCEEDS MMO STORAGE NEEDS

Companies providing MMO back-end services clearly need low latency storage that performs like local NVMe flash, but is centralized, shared, and supports data services such as thin provisioning. Because deployments are often in edge data centers with limited hardware choices, must make use of existing servers, or simply cannot support special NICs in servers, NVMe over Fabrics solutions based on RDMA over Converged Ethernet (RoCE) are often simply unacceptable.

TCP, meanwhile, is ubiquitous. Proprietary arrays are undesirable and may not be able to be deployed in all necessary geographic locations. Until recently, no storage solution could meet all of these challenges - but now one can: Lightbits Labs LightOS™.

LightOS is software-defined NVMe/TCP block storage that exceeds the requirements for scalable, resilient and low latency pooled NVMe for online gaming back-ends. Reviewing the requirements from the previous sections, we can see how and why this is the ideal solution for MMO storage needs.

ENVIRONMENT

Most MMO backends make extensive use of Linux, and utilize bare-metal, Openstack or Kubernetes application environments. NVMe/TCP client drivers are included or readily available for nearly any modern Linux kernel and distributions. LightOS NVMe/TCP target software can integrate into bare-metal, Openstack (via Cinder driver) or Kubernetes (via CSI Driver) environments. Similar to Kubernetes failure domain functionality, LightOS clustering supports the notion of failure domains for power zone, rack-level or row-level failures - while continuing to seamlessly serve storage.

PROVISIONING, MANAGEMENT AND MONITORING

LightOS is managed via a robust, RESTful API and command line interfaces for integration into various frameworks and/or scripting environments. The software itself is delivered as packages and utilizes Ansible for installation and configuration. For monitoring, diagnostics and telemetry, LightOS utilizes Prometheus and comes with pre-configured Grafana dashboards that can be used as is, or as an example for inclusion in customized monitoring environments.

LightOS integrates into your environment with modern, efficient interfaces, methods and practices.

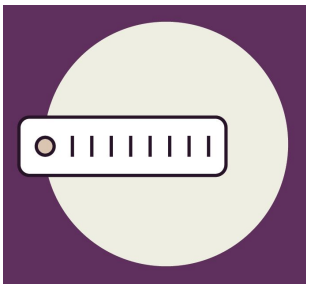
APPLICATION BENEFITS

LightOS excels at providing low-latency, high IOPs storage service for transactional loads such as databases and message streaming services such as those utilized in gaming backends. In best practices documentation for nearly all modern, cloud native databases and Apache Kafka, utilizing local flash, (especially NVMe), in a DAS fashion is suggested. This leads to severe under-utilization, administrative silos, and physical dependencies around what applications can run on which servers.

LightOS disaggregated NVMe/TCP block storage allows for maximum utilization and flexibility while preserving local flash performance. It eliminates application to server binding by allowing applications to run on any server with TCP/IP connectivity in the data center. Logical volumes can be any size and grown if needed, freeing applications from physical drive constraints.

FLEXIBILITY

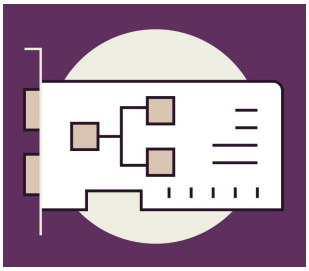
As software, LightOS can run on a practically infinite number of platform configurations. As a disaggregated solution, it allows for independent scaling of compute and storage. It also allows for the re-use of application servers without special NICs and extends their useful life by eliminating a service element (drives) and allowing for logical volume growth without physical server changes.



Standard x86 servers

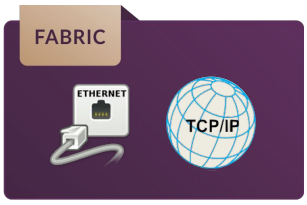
Additionally, this flexibility of deployment makes it ideal for edge data center locations where limitations on equipment, power and cooling, or the inability to configure Ethernet switches for special protocols might limit proprietary hardware solutions or network protocols such as RoCE.

LightOS is perfect for addressing the needs of burst usage, with tens of millions of IOPs capability in a single cluster and the ability to scale. Thin provisioning ensures the ability to provision terabytes of volumes on the fly, but only actual written blocks will consume precious flash media.



Any Ethernet NIC

Additional NVMe drives can be added to LightOS target servers at any time assuming there are empty slots available. These features allow for maximum utilization and can potentially save significant costs versus thick provisioning storage in the cloud. Once a temporary need is finished, that space is simply and quickly returned to the NVMe pool.



Standard TCP/IP networks

Lastly, LightOS helps future-proof deployments by allowing any server with any Ethernet NIC to access shared NVMe storage. It scales when needed, and supports online, rolling upgrades so your storage stays up while the cluster is upgraded with new features and functionality.

LOGGING AND TELEMETRY

Real-time analytics, debug logging and telemetry need consistent, low-latency storage performance and LightOS delivers. LightOS specializes in low-latency performance with tail latencies that are better than local NVMe DAS drives. At the same time, special post-processing analytics can require high bandwidth. LightOS delivers that as well.

CONCLUSION

Online gaming support infrastructure must contend with a frustrating enemy: lag. To combat lag, back-end gaming infrastructures must eliminate latency wherever possible to avoid players experiencing lag caused by that back-end. As storage is the bottommost layer, any latency ripples through the upper layers and can have a serious effect on a player's experience.

LightOS is a software-defined, scalable shared NVMe block storage solution that can be a serious weapon in the fight against lag. Low-latency, high utilization and flexibility at lower TCO than DAS can propel your MMO back end to new lows, when it comes to latency.

FIND OUT MORE

To learn more, please visit our website, www.lightbitslabs.com
To contact our team, email us at info@lightbitslabs.com

LightOS™