Apache Kafka and LightOS™

Introduction

Apache Kafka was originally developed by LinkedIn and was open sourced in early 2011. The original use case for Kafka was in the area of website activity tracking. LinkedIn wanted a way to track user activity in real-time and needed a service that could take in events from various services ("producers" in Kafka speak) and be consumed by other services ("consumers") in order to have a more dynamic and immersive end-user experience. For this to work, Kafka had to be scalable, low latency and highly available.

It wasn't enough for events to be exchanged in real-time between various systems; there was also a desire to feed data into Hadoop, or to data warehousing systems for later, offline processing and reporting. This need to exchange data with external systems, possibly at a later date, meant Kafka data had to be durable. Thus durability could not make performance suffer though - the system had to perform with low latency regardless of the amount of historical data stored.

Today, Kafka is used by such well-known sites as LinkedIn, Twitter and Netflix in addition to thousands of other companies world-wide. Use cases include:

- **Website activity tracking**
- **Metrics/monitoring**
- **Real-time Analytics**
- **Log aggregation**
- **Stream processing**
- **Distributed commit logs**
- **Fraud detection**

While every use case is different, they all share a need for low-latency, reliability and scalability.

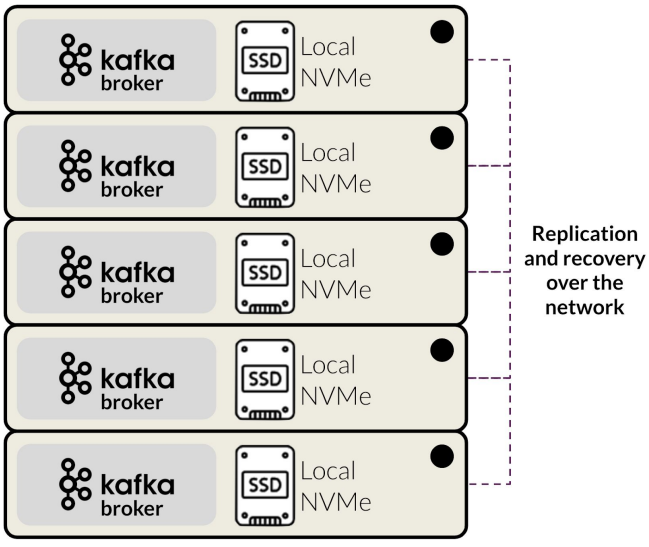
Kafka Deployment

Kafka is *unlike* most messaging systems in that the message log is always persistent. That is, messages are immediately written to a filesystem when they are received.

To ensure the storage can keep up with these real-time needs, Kafka best practices suggest using local NVMe SSD drives with a file system per drive and a host of rules, restrictions and trade-offs.

Many of these trade-offs are minor, but some hinder flexibility and force decisions that directly affect Kafka configuration and use.

Lightbits Labs believes your storage choice should enhance your applications - not dictate their behavior.



Typical Apache Kafka cluster server deployment

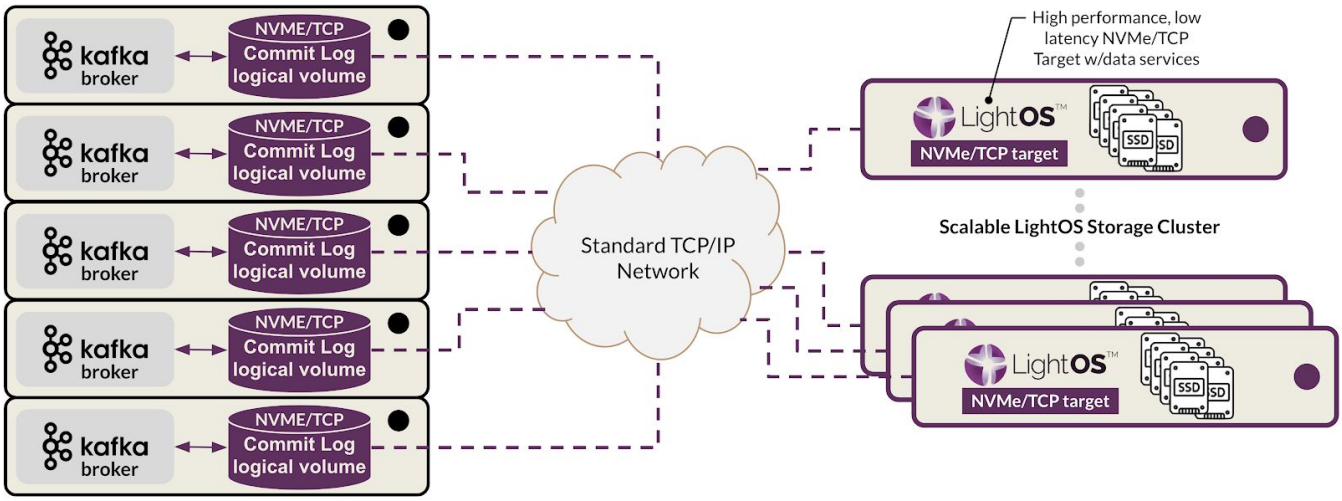
© 2020 Lightbits Labs™ Proprietary and Confidential

LightOS and Apache Kafka Together

LightOS, by Lightbits Labs is a software defined storage product with a similar philosophy as the Apache Kafka project:

- Storage must be consistent, low-latency and perform at the speed of NVMe SSDs
- Communication between application servers and LightOS target servers is done with a simple, high-performance, and standard TCP protocol - NVMe/TCP
- Storage must be scalable, highly available and perform consistently
- Storage consumers must be able to make a choice on data protection level and/or response time

By replacing local NVMe SSDs on Kafka servers (brokers) with logical volumes provided by LightOS via NVMe/TCP, Kafka becomes an even more powerful tool. Utilizing LightOS volumes in place of local NVMe drives enhances flexibility, performs like local NVMe and yet removes many of the trade-offs of direct attached flash.



Apache Kafka cluster server deployment enhanced with LightOS NVMe/TCP storage

Ensuring Data Durability

The Reliability vs. Performance Trade-off

To ensure reliability, Kafka allows producers to wait on acknowledgement so that a write isn't considered complete until it is fully replicated and guaranteed to persist even if the server it was written to fails[1]. Most environments leave the default setting of replication set to '3'. Extra replicas take time, and can also affect consistency of writes unpredictably. Producers have the option of either waiting for the message to be committed or not, depending on their preference for tradeoff between latency and durability[2]. This preference is controlled by the "acks" setting that the producer uses.

LightOS ensures consistent write response as the block devices are completely virtualized and the intelligent flash management layer chooses data placement of the underlying NVMe media in part based on drive behavior. SSDs going into "garbage collection" mode, will not adversely affect Kafka brokers as they would if they are local to the Kafka server. Thus, if maximum reliability is desirable with replicas set to '3', this can be achieved with much more consistent response. Alternatively, because LightOS is protecting against drive failures with Elastic RAID, replication can be set to '2' ensuring lower write latency while still ensuring protection against a Kafka server failure.

© 2020 Lightbits Labs™ Proprietary and Confidential

Better Protection Against Drive Failures

In a legacy Kafka deployment, a drive failure would result in a full data rebuild by the surviving replicas on other servers, over the network. This can have a long and detrimental effect on the network and the Kafka service itself.

LightOS protects against drive failures with Elastic RAID on the LightOS target hosts themselves. In the case of a drive failure, any data written to the drive will be seamlessly rebuilt into spare capacity within the target server with minimal service disruption. This means the recovery takes much less time and does not impact network performance or burden surviving replicas on other brokers with additional load.

Improved Availability and Flexibility

Avoiding Synchronization Issues

For a given topic in Kafka, the “leader” is responsible for taking the message from a producer and replicating it to “followers”. A follower must replicate the writes happening on the leader and not fall "too far" behind[2]. If it does fall behind, this might lead to declarations of out-of-sync replicas and could affect incoming producers depending on the replication policy and the producers’ “acks” setting.

LightOS separates read and write paths in its intelligent flash management layer and offers consistent response time for reads and writes with lower tail latencies than local NVMe drives. This consistency extends to all brokers. With local SSDs, a single drive becoming slow or unresponsive on a broker acting as a follower can cause that follower to fall behind. This condition is far less likely with LightOS, avoiding temporary and/or flapping “in-sync” conditions.

Avoid Drives Dictating Data Placement

When deploying Kafka with local SSDs, topic partitions are bound by the capacity limit of a given drive. If you want to use multiple SSDs without RAID in a Kafka server you can do so with each being a different data directory and partitions will be assigned round-robin to data directories. Each partition will be entirely in one of the data directories. If data is not well balanced among partitions this can lead to load imbalance between SSDs

With LightOS, virtual volumes made available to Kafka servers can be of arbitrary size. They can be later expanded online if data retention needs for a given topic and partition change.

Avoid Full Rebuilds Due To Kafka Server Failures or Migrations

As a shared nothing architecture with local SSDs, a Kafka server failure results in the need for a full rebuild of the data on any local SSDs. Additionally, a planned migration (to perhaps, a more powerful server) requires adding the new server to the cluster and decommissioning the old server, resulting in a full network synchronization.

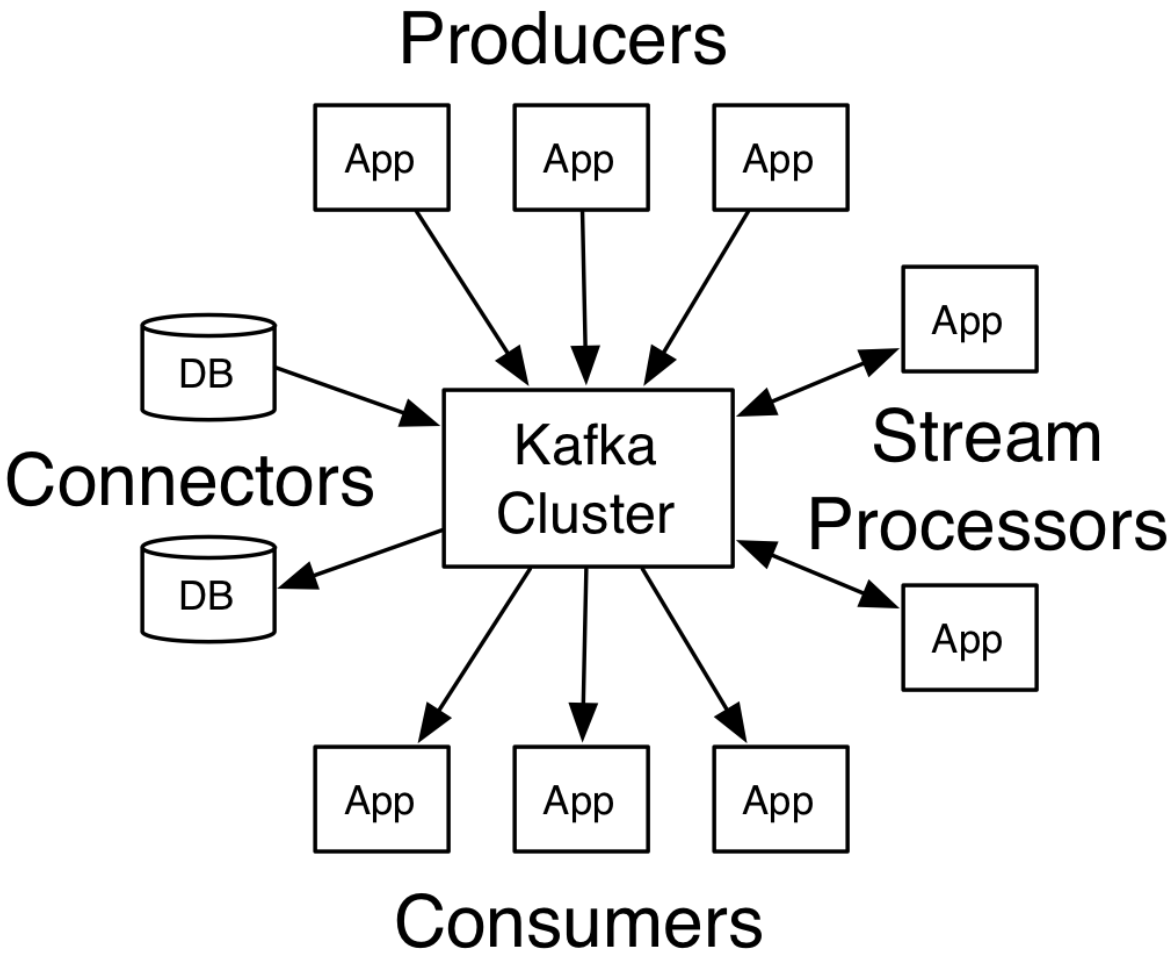
LightOS can avoid this full synchronization/rebuild by virtually moving the logical volume to a replacement/new server that utilizes the old server’s ID. Once up, only writes that were missed during the brief downtime need to be synchronized.

Similarly, LightOS virtual volumes allow for independent scaling of Kafka compute and storage meaning new Kafka nodes can be added at will without local NVMe SSD if more message handling power is needed. Conversely, if just more capacity is needed NVMe drives can be added to existing nodes and volumes can be expanded without disruption, or new LightOS targets can be added to the cluster for additional capacity.

© 2020 Lightbits Labs™ Proprietary and Confidential

Deploying Kafka In Containers and/or With Connectors

Kafka is increasingly "in the middle" of an entire ecosystem of producers, consumers, connectors and applications:



It's not uncommon for Kafka to be taking in messages from various producers and sending data via connectors into Apache Spark or Cassandra.

In addition to various applications, Kafka is often deployed in containers. Entire workflows are being packed into orchestration and application stacks. One stack, called SMACK, combines Apache Spark, Apache Mesos, Akka, Cassandra, and Kafka to implement a type of CQRS (command query responsibility separation). This stack benefits from powerful ingestion (Kafka), back-end storage for write-intensive apps (Cassandra), and replication to a more query-intensive set of apps (Cassandra again). All of this can be managed with a resource/cluster management solution such as Apache Mesos.

In a complete stack or an arbitrary application ecosystem, these applications need persistent storage. In a container environment such as Mesos or Kubernetes, LightOS provides a fully integrated persistent storage solution using the CSI interface. Of course, in bare metal environments, LightOS performance, composability and rich features make it ideally suited to any transactional workload requiring low latency and high availability.

© 2020 Lightbits Labs™ Proprietary and Confidential

Enhancing Performance

Memory and Caching

In Kafka's efforts to keep things simple, an architectural decision was made to rely on the Linux file system's tendency to cache up to the limits of available memory. Within Kafka, all data is immediately written to a persistent log on the filesystem without necessarily flushing to disk[3]. In effect this just means that it is transferred into the kernel's pagecache. If we examine this behavior in the case of reads, requests for historical records or any messages that are not cached become 100% dependent on the performance of the underlying storage devices. Thus, response time and the consistency of that response become paramount for mission critical environments.

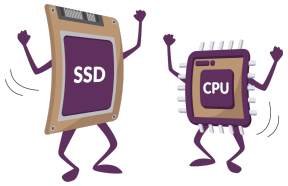
As detailed above, LightOS provides consistent low latency, regardless of a mix of simultaneous reads and writes. In cases such as these, LightOS can perform better than local NVMe SSDs (via more consistent response times), especially in light of high incoming message rates on a given Kafka broker.

Conclusion

The Lightbits LightOS software defined storage solution enables separation of storage and compute for Kafka workloads. It reaches the same transactional performance as local NVMe SSDs, despite I/O going over your standard data center network. It supercharges your Kafka applications while increasing reliability and flexibility by providing:

- The same performance as DAS with greater utilization of your storage investment
- Improved service levels and a better user experience with consistent latency
- Faster rebuild time with higher resiliency levels
- Standard, simple, secure storage access to any of your application servers
- No changes to your TCP/IP network with no changes to application servers

No more managing a growing number of Kafka servers with their own underutilized silos of direct attached storage inhibiting simple and efficient cloud-scale architecture.



Finally, you can separate storage from compute without all the drama, and at lower cost.

For more information about the Lightbits solution, contact info@lightbitslabs.com.

© 2020 Lightbits Labs™ Proprietary and Confidential

References

1. Apache Kafka Introduction: Kafka as a Storage System - https://kafka.apache.org/intro
2. Apache Kafka Documentation: 4.7 Replication - https://kafka.apache.org/documentation/#replication
3. Apache Kafka Documentation: 4.2 Persistence: Don't fear the file system - https://kafka.apache.org/documentation/#design_filesystem

About Lightbits Labs™

The Lightbits NVMe/TCP solution is the latest innovation from the Lightbits team, who were key contributors to the NVMe standard and among the originators of NVMe over Fabrics (NVMe-oF). Unlike other NVMe-oF approaches, the Lightbits NVMe/TCP solution separates storage and compute utilizing standard drivers and TCP/IP network protocol. With NVMe/TCP, Lightbits delivers the same IOPS as direct-attached NVMe SSDs and up to a 50% reduction in tail latency. The transition is so smooth your applications teams won't even notice the change. Once they switch to NVMe/TCP, they can scale storage and compute independently and with consistently better user experience.

www.lightbitslabs.com info@lightbitslabs.com

US Office
1830 The Alameda,
San Jose, CA 95126, USA

Israel Offices
17 Atir Yeda Street,
Kfar Saba, Israel 4464313

3 Habankim Street,
Haifa, Israel 3326115

The information in this document and any document referenced herein is provided for informational purposes only, is provided as is and with all faults and cannot be understood as substituting for customized service and information that might be developed by Lightbits Labs ltd for a particular user based upon that user's particular environment. Reliance upon this document and any document referenced herein is at the user's own risk.

All product and company names are trademarks™ or registered® trademarks of their respective holders. Use of them does not imply any affiliation with or endorsement by them.

The software is provided "As is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall the contributors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings with the software. Unauthorized copying or distributing of included software files, via any medium is strictly prohibited.
COPYRIGHT (C) 2020 LIGHTBITS LABS LTD. - ALL RIGHTS RESERVED