Virtual IMS user group: Newsletter 67

Welcome to the Virtual IMS user group newsletter. The Virtual IMS user group at www.fundi.com/virtualims is an independently-operated vendor-neutral site run by and for the IMS user community.

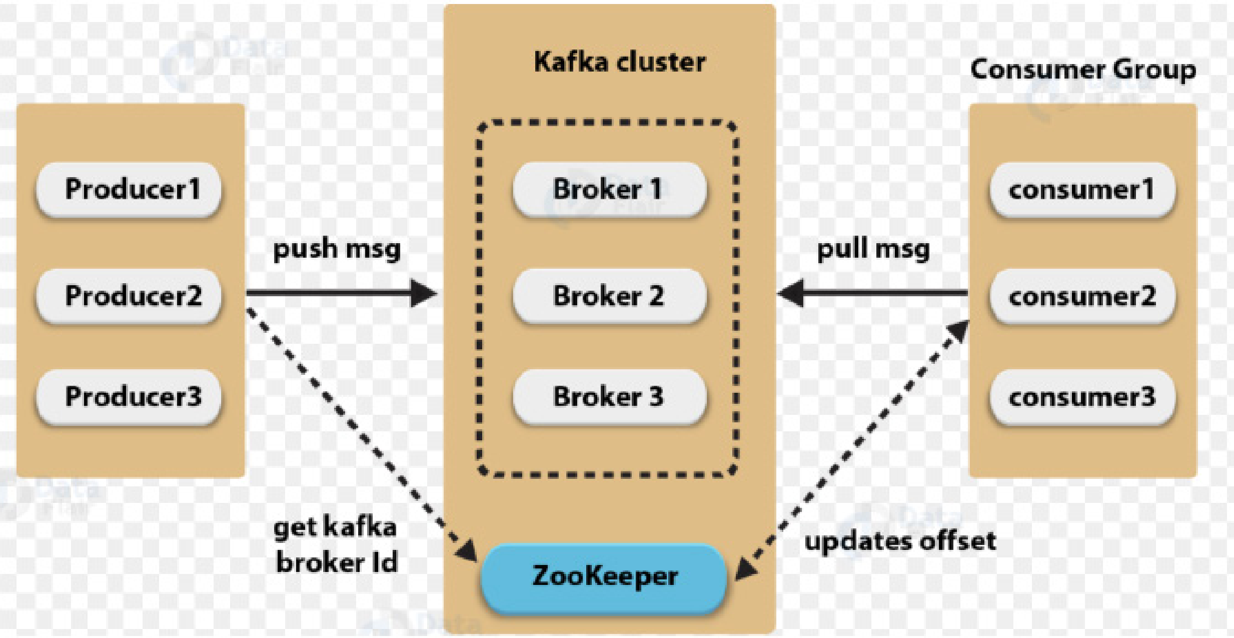

Figure 1: Kafka model

Virtual IMS user group presentation

The latest webinar from the Virtual IMS user group was entitled, “IMS CDC to Kafka – Performance and Tuning”. It was presented by Scott Quillicy, Principal at SQData Corporation.

Scott is a co-founder of SQData, a US-based independent software vendor (ISV) that specializes in high-performance data

Contents:

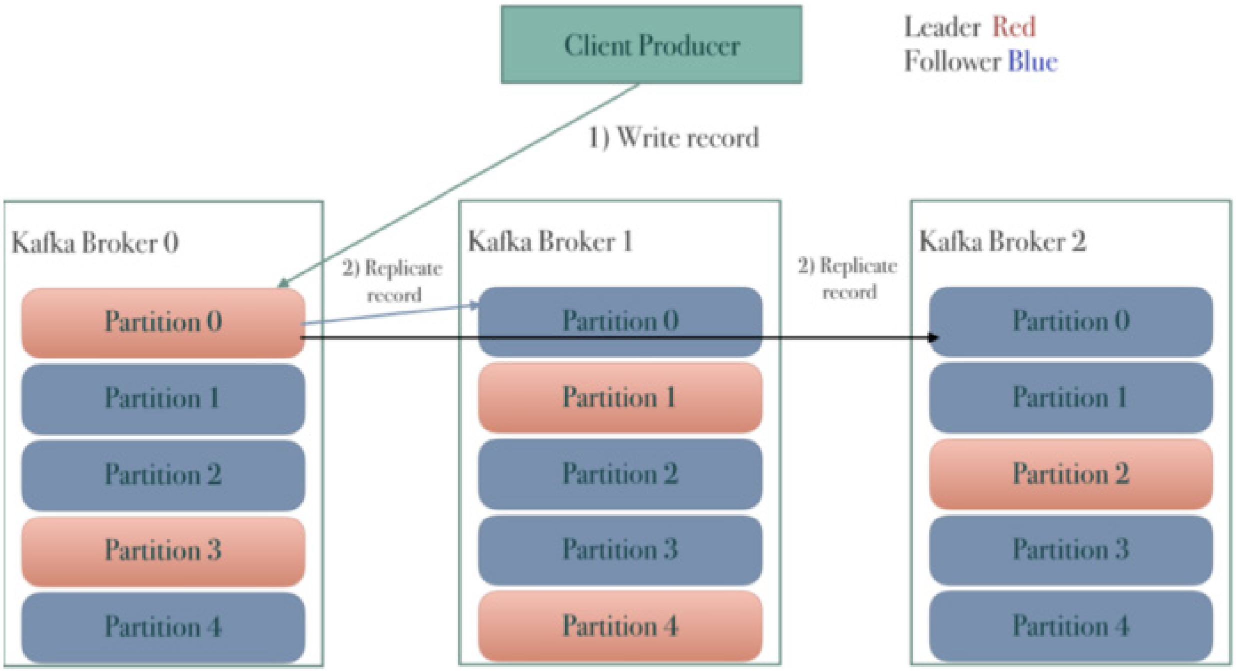

Figure 2: Kafka replication

replication and changed data capture (CDC) solutions for mainframe (IMS, VSAM, DB2) and distributed databases. With almost 40 years of experience, Scott is considered by his peers to be an expert in database replication technology and deployment. Over the past few years, Scott has been working with larger mainframe customers on their real-time data streaming initiatives, most of which involve high-volume ingestion of operational data into fast targets such as Kafka and Hadoop using SQData's mainframe CDC connector technology.

Scott Quillicy started his presentation by looking at what Kafka is (see Figure 1). He said that it was an ultra-fast distributed stream processing platform for big data. It's open source with a very large community. It's a publisher / subscribe messaging system that's highly scalable, durable, and fault-tolerant. Topics (queues) and topic partitions are separate log files.

In Kafka, a record is considered 'committed' when all ISRs (in-sync replicas) for a partition have written to their logs. Only committed records are readable by consumers.

Scott looked at IMS transactions and Kafka's commit scope, saying that it was an interesting mix of ACID (Atomicity, Consistency, Isolation, Durability) and BASE (Basic Availability, Soft state, and Eventual consistency). The source IMS is ACID, using commits and rollbacks, whereas the target Kafka is currently BASE, ie no commits or rollbacks.

For Kafka, each message is acknowledged. It's not transaction-based, each topic is acknowledged independently. You can use the "At Most Once" option, which is fastest, but you could lose data; or "At Least Once, which is durable, but could lead to duplicates; or "Exactly Once", where messages are logged exactly once, so there are no duplicates, but it does have some additional

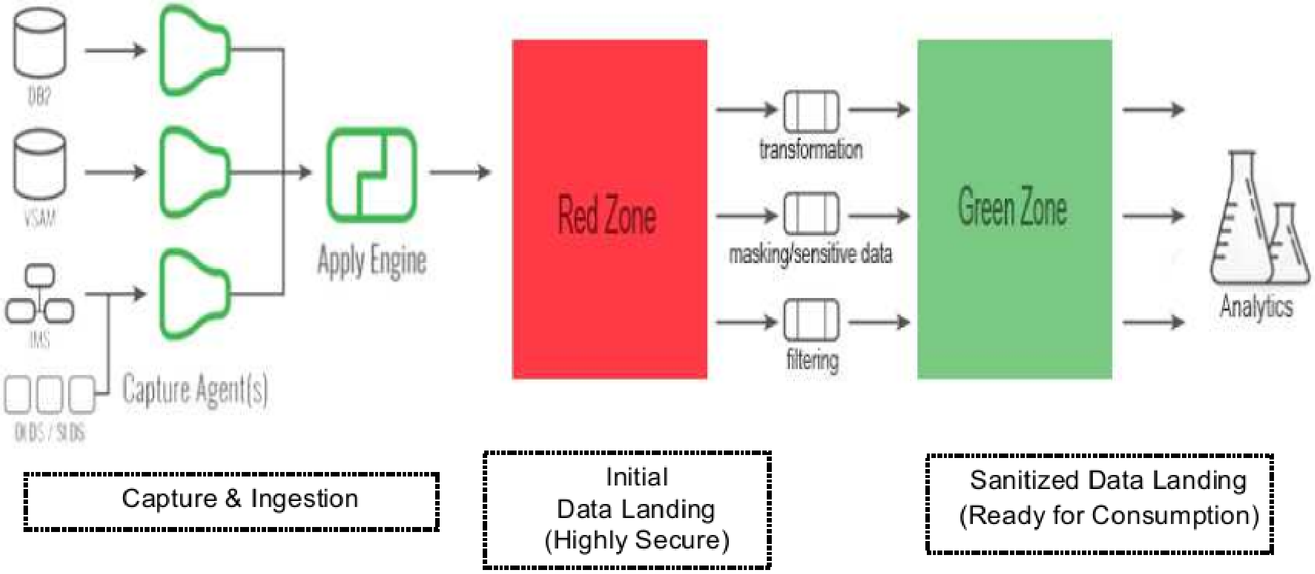

Figure 3: Common CDC streaming deployment model

overhead. In real life, the commit scope is based on business requirements. It's worth planning for duplicate messages. Kafka replication is shown in Figure 2.

Figures 3 shows the common CDC streaming deployment model.

The key thing with performance is that throughput / latency primarily depend on the speed of the target – and Kafka is a very fast target. The top items affecting performance / throughput are: message size, component scaling factor, IMS initial load configuration and data volume, IMS CDC streaming configuration (track size and arrival rate), and Kafka configuration.

Small messages perform better than larger ones (obviously!), so it boils down to the number of bytes per message. Of course, IMS segments can be large, and updates can contain before and after images. So, what can you do? Use Avro as the target data format. It's a condensed version of JSON. JSON is typically used for data validation only. Avro messages are roughly the size of a source segment. You can reduce the number of fields in the CDC message. And you can use compression. This reduces the footprint on disk and on the network. lz4 appears to be most stable method.

The initial load is illustrated in Figure 4. The initial load is required to prime the target(s). You can load using mass inserts; there are no 'traditional' utilities on the target side. An alternative is to use online snapshots; ie FTP, transform, ingest into Kafka. You need to be able to scale on both the source and the target sides. Importantly, you need to be able to run unloads against live databases. Scott recommended using a separate set of topics for the initial load.

Scott Quillicy then went on to look at scaling issues on the target side and the source side.

Scott then discussed Kafka topic partitions, suggesting that they increase throughput through parallelism and are highly recommended. On

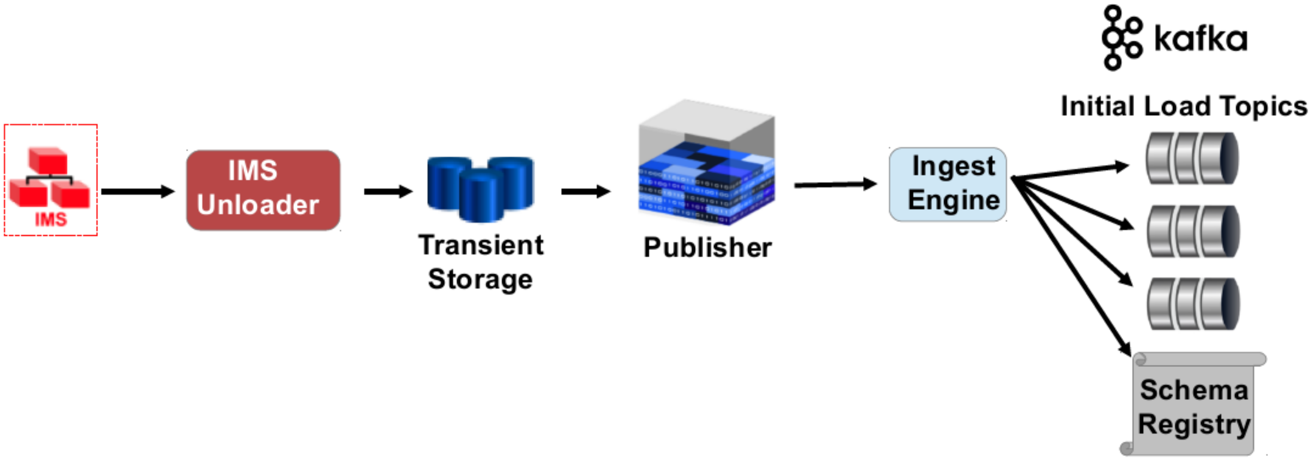

Figure 4: The initial load

the downside, they increase the chances of unavailability if a broker abends. For example, it takes longer to assign a new leader if there are many partitions. As a rule-of-thumb, it takes 5ms per topic partition, so 10,000 partitions will take about 50 seconds. The other negative effect is that it may increase end-to-end latency. Consumers can only see messages that have been fully replicated. It takes about 20ms to replicate 1,000 partitions between brokers. This is somewhat alleviated if running a large cluster. Scott's recommendation was 100 x b x r, where b is the number of brokers and r is the replica factor.

Scott's final tips were:

- To keep the ingestion method as simple and consistent as possible.
- To understand the business requirements, eg end-to-end latency expectations, and the fact that near real time may require more scaling.
- Scale to satisfy business requirements, don't overdo it.
- Have a solid recovery plan.
- Be patient with 'The Great Divide' between mainframe ways of working and Kafka-centric staff.
- Select the right tools for the job.

A copy of Scott Quillicy's presentation is available for download from the Virtual IMS user group Web site at www.fundi.com/virtualims/presentations/IMSCDCtoKafkaApr19.pdf.

You can see and hear the whole user group meeting at https://youtu.be/hQMqYSi7P-k.

Meeting dates

- On 11 June 2019, James Martin, Senior Solutions Advisor for Rocket Software, will be discussing "What's new in IMS Connect Extensions V3.1".
- The following meeting will be on 6 August when Dusty Rivers, Director, z Systems Software: IMS & CICS at GT Software, will be speaking.

Recent IMS articles

Get IMS and the cloud to your z/OS environment by Dipti Gandhi on z Systems

Developer Community (4 April 2019). You can find the article at https://developer.ibm.com/zsystems/2019/04/04/get-ims-and-the-cloud-to-your-z-os-environment/

Sponsorship opportunity

Are you missing a great opportunity to advertise your IMS software?

The Virtual IMS user group is now offering software vendors the opportunity to advertise their products in a number of ways. You could have an advert on the home page of the Web site (at www.fundi.com/virtualims), you could advertise in the newsletter, and/or you could advertise in the monthly e-mails sent to members of the user group.

E-mail trevor@itech-ed.com for full information about marketing opportunities with the Virtual IMS user group.

Like us on Facebook

#VirtualIMS

About the Virtual IMS user group

The Virtual IMS user group was established as a way for individuals using IBM's IMS hierarchical database and transaction processing systems to exchange information, learn new techniques, and advance their skills with the product

The Web site at www.fundi.com/virtualims provides a central point for coordinating periodic meetings (which contain technically-oriented topics presented in a webinar format), and provides articles, discussions, links, and other resources of interest to IBM IMS practitioners. Anyone with an interest in IMS is welcome to join the Virtual IMS user group and share in the knowledge exchange.

To share ideas, and for further information, contact trevor@itech-ed.com.

The Virtual IMS user group is free to its members.