Nallatech
a molex company

White Paper

Any Rate, Any Format: Accelerating Kafka Producers with FPGAs

Introduction

Apache Kafka is at the heart of emerging universal streaming data pipeline. Kafka's has many high-profile adoptions as the streaming platform of choice being used at LinkedIn, Netflix, Uber, ING along with over one third of the Fortune 500 and growing. At LinkedIn, approximately two trillion messages per day pass through Kafka. According to TechRepublic.com, six of top 10 travel companies, seven of top 10 global banks, eight of the top 10 insurance companies and nine of top 10 telecom companies have adopted Kafka as the central platform for managing streaming data. At the 2017 New York Kafka Summit, Confluent reported over one third of the Fortune 500 have deployed Kafka.

Basic Kafka System

Kafka has three essential components – producers, brokers and consumers. Producers publish data to topics on brokers and consumers subscribe to topics. Figure 1 shows a basic Kafka system.

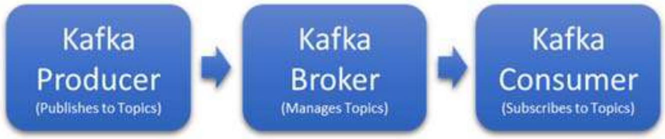

Figure 1 - Basic Kafka System

One of many the advantages of the Kafka architecture is the decoupling of producers and consumers. Producers and consumers can be at wildly different data rates and yet have no effect on each other. The other key advantage of Kafka is its small size. With just over 90,000 lines of code, Kafka clusters can be implemented on much more modest hardware requirements than Spark Streaming which requires a full Spark node.

Accelerating Kafka Producers

Data ingest into big data systems ranges from simple to complex. In figure 2, data source 1 may be a packet captures of network traffic. However, data source two could be complex geospatial images from a constellation of satellites, while data source three is industrial IoT maintenance data on a windmill farm in West Texas.

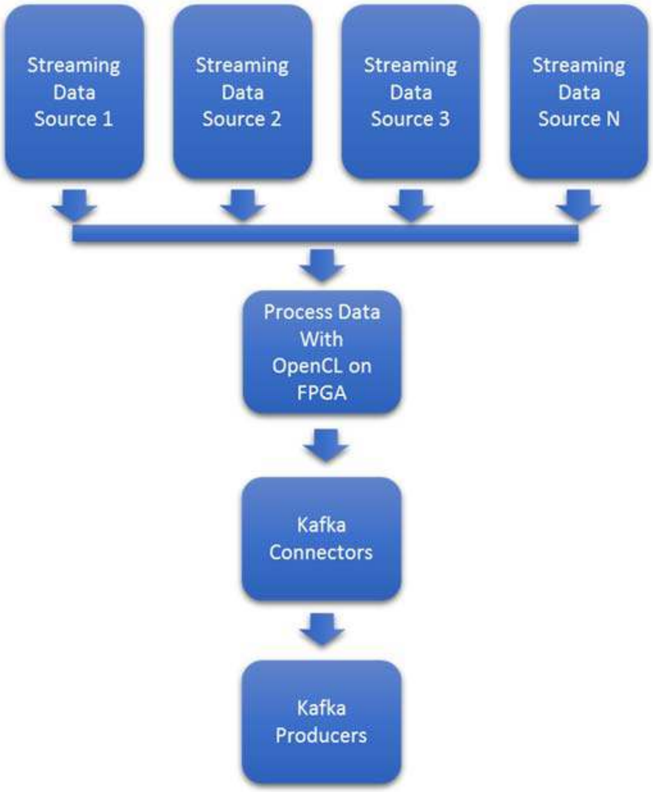

Figure 2 : Streaming Data Ingest Acceleration with Intel FPGAs

The variability in data formats and data rates makes the problem difficult to scale. Being able to adapt in real-time to burst in traffic and new formats is often costly requiring provisioning of additional NICs and processors. Figure 3 shows a typical processor based architecture used in most Kafka clusters.

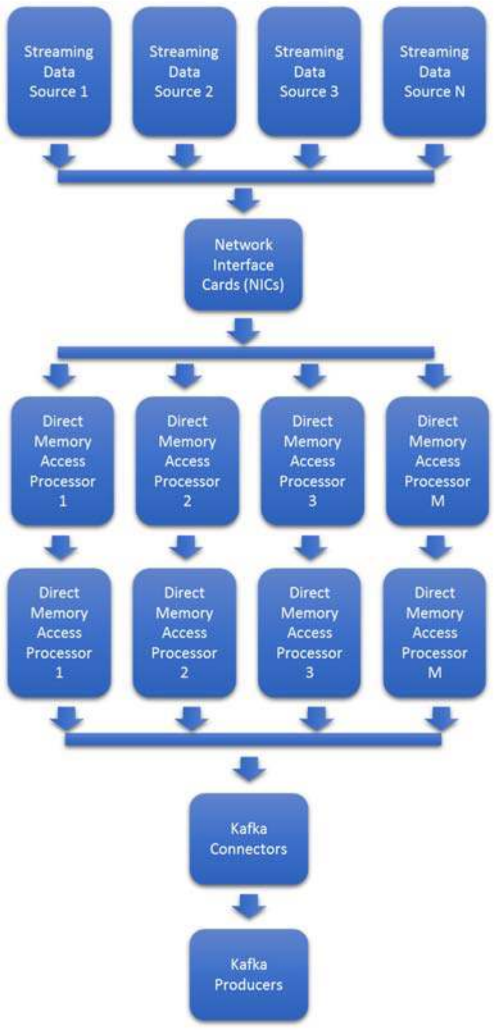
Streaming Data Source 1
Streaming Data Source 2
Streaming Data Source 3
Streaming Data Source N
Network Interface Cards (NICs)
Direct Memory Access Processor 1
Direct Memory Access Processor 2
Direct Memory Access Processor 3
Direct Memory Access Processor M
Direct Memory Access Processor 1
Direct Memory Access Processor 2
Direct Memory Access Processor 3
Direct Memory Access Processor M
Kafka Connectors
Kafka Producers

Figure 3 : Typical Ingest Pathway

Data rate variability makes the system in figure 3 difficult to plan. In many cases, the maximum bandwidth must be estimated and then provisioned. 50% or more excess processors and NICs will be idled waiting for increases in data rates.

Moving to an Intel FPGA based solution, the same maximum bandwidth will be estimated, but the simplified system in figure 4 will have much lower power while idle and requires considerable less footprint overall. The system in figure 2 will also eliminate flow control and load balance management needed for processor based system because the Intel FPGA based approach is deterministic regardless of data rate or data formats.

Intel FPGAs are streaming, parallel accelerators that attach directly to copper, fiber & optical wires. Unlike traditional GPUs and CPUs, Intel FPGAs can move any data in any format from wire to memory in nanoseconds without the need of a Network Interface Card (NIC).

This acceleration of ingest can result in 40X lower latency in data ingest to Kafka producer. It provides the option for simultaneous real-time processing of the inflowing data such as by implementing machine learning, image recognition, pattern matching, filtering, compression, encryption etc. Ingested data can be therefore accelerated and enriched to speed time to data acquisition and data analysis.

Use Case One:
Inline Extract & Transformation

The most basic use case for FPGA ingest into a Kafka producer is shown in figure 4. Even for this most basic use case, the FPGA provides low latency and determinism for even extremely variable rates. The ability to extract and transform the data with OpenCL allows this use case to handle 10s to 100s of data types.

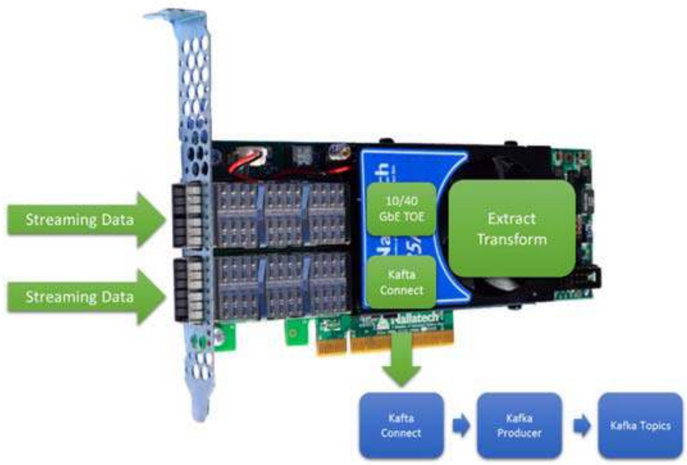

Figure 4 Inline, Low Latency, Deterministic, Extraction & Transformation

Use Case Two:
Inline Encryption & Decryption

Encryption is extremely expensive in processor cycle, but well understood on Intel FPGAs. FPGAs provide a low latency and deterministic result without a dependency on the data rate. For processors, variable data rates could flood the processor resources and cause a bottleneck and/or start dropping packet.

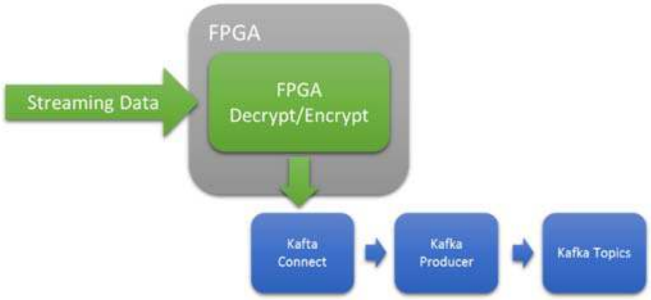

Figure 5 Inline, Low Latency, Deterministic, Encryption or Decryption

Use Case Three: Inline Compression & Decompression

FPGA's are extremely efficient at compression and decompression. In this use case the FPGA is used to compress/decompress data before it is passed to the Kafka system.

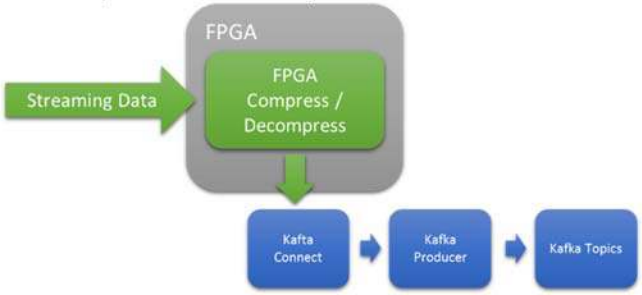

Figure 6 Inline, Low Latency, Deterministic Compression or Decompression

Use Case Four: Information Theory with Encrypted/Decrypted & Compressed/Decompressed Streams

Shannon's law is being applied to more streaming use cases to determine if a stream is encrypted. Shannon's law calculates the entropy of a packets looking for randomness versus structured bytes. Many encrypted bytes look, but not all, similar structured data. Figure 7 shows a possible flow to calculate the entropy, attempt to decrypt and then decompress before being published to a Kafka topic. Even if the decryption and/or decompress could not be done successfully, sorting encrypted vs decrypted streams has many applications in industries, such as personal identifiable information like finance and health care.

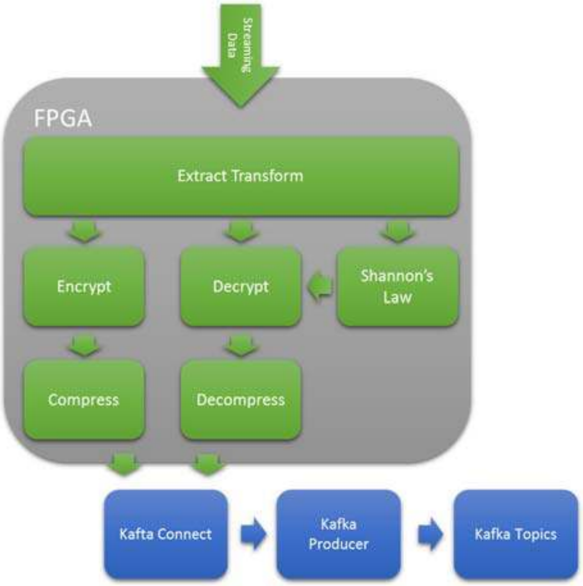

Figure 7 Inline, Low Latency, Deterministic, Complex Information Theory Processing

Use Case Five: Enriched Topic Routing

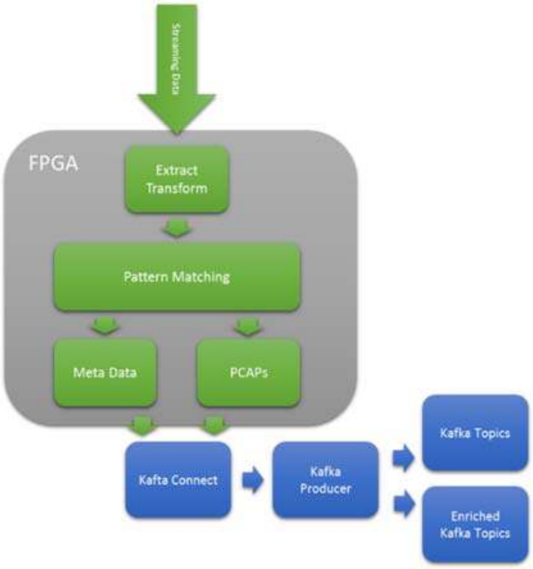

Figure 8 Enriched Topic Routing of PCAPs for Cyber Analytics

Kafka's flexible topic architecture that allows ingested data to be placed into many topics. This flexibility means incoming data can be routed/switched using machine learning and pattern matching. Take figure 9 above which shows raw network packets being captured (PCAPS). As the packets are captured, complex pattern matching using PCRE expressions can route to the appropriate topics. This allows the Kafka consumers to subscribe to enriched topics and bypass a cleaning stage. For many cyber analytics applications, the processing realizes a 1000X improvement in cyber operations per watt based on research published by DOE Sandia & Lewis Rhodes Labs.

Nallatech 385A Cloudera/Intel Example

The Nallatech 385A provides two network ports supporting up to 40Gbe/sec each. This NIC size card can replace existing NIC/CPU combination to significantly accelerate existing Kafka networks and reduce power.

This has been verified by Cloudera and Intel to accelerate Kafka to Spark streaming, whilst performing data enrichment on the FPGA (Figure 9).

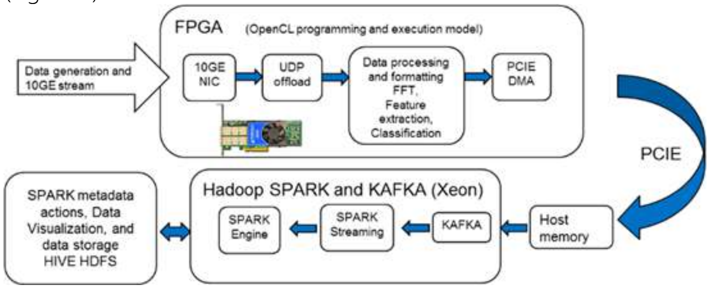
FPGA
(OpenCL programming and execution model)
Data generation and 10GE stream
10GE NIC
UDP offload
Data processing and formatting FFT, Feature extraction, Classification
PCIE DMA
PCIE
SPARK metadata actions, Data Visualization, and data storage HIVE HDFS
Hadoop SPARK and KAFKA (Xeon)
SPARK Engine
SPARK Streaming
KAFKA
Host memory

Figure 9 Enriched data using 385A

In the above demonstration, we have chosen engine noise signatures as our input data stream. They are ingested and offloaded via an UDP offload engine and placed into the card's OpenCL environment. OpenCL code running on the card performs real-time formatting on the incoming data stream. It then performs an FFT, feature extraction and classifies the signal as "normal" or "abnormal" based on comparison with known engine signatures. This extra bit of data along with the FFT of the engine signals are DMA into Kafka for further processing.

This example also highlights the flexibility of OpenCL generated libraries which can be applied to incoming streaming data. This offers then end user immense latitude to include very application specific forms of data enrichment or data filtering.

520N: 100 Gbe with Stratix10

The Nallatech 520N four network ports enable support for an array of serial I/O protocols operating up at 10/25/40/100Gz. With a total throughput of up to 400 Gbe/sec, the 520N is cable of enriching high volumes of data prior to offloading to a Kafka framework.

Figure 10 Enriched data using 520N

The 520N is populated with the powerful Stratix 10 FPGA offering unparalleled performance.

With the combination of high throughput, large amounts of compute and programmability using OpenCL, it is possible to perform complex data enrichment on streaming data on a single device.

More Information and How to Evaluate

Nallatech along with Intel PSG are experts at Kafka acceleration. Nallatech has current and planned products to accelerate Apache Kafka using Arria 10 and Stratix 10 FPGAs. Please contact Nallatech to discuss your needs and develop an accelerated solution.

Contact Details

741 Flynn Road
Camarillo, CA 93012, USA

Tel: (805) 383-8997

Toll Free: 1-877-44-NALLA

Fax: (805) 384-4899

1 Napier Park
Cumbernauld, G68 0BH, UK

Tel: +44 (0)1236 789567

Fax: +44 (0)1236 789599

contact@nallatech.com

Visit us online: www.nallatech.com

Any Rate, Any Format:
Accelerating Kafka Producers with FPGAs

White Paper

United States office:
741 Flynn Road, Camarillo, CA 93012, USA

Tel: (805) 383-8997 Toll Free: 1-877-44-NALLA Fax: (805) 384-4899

UK office:
1 Napier Park, Cumbernauld, G68 0BH, UK
Tel: +44 (0)1236 789567 Fax: +44 (0)1236 789599