Amazon Kinesis

ABHISHEK GUPTA[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: abhigupt@iu.edu*

project-001, April 8, 2017

Amazon Kinesis [1] provides a software-as-a-service(SAAS) platform for application developers working on Amazon Web Services(AWS) [2] platform. Kinesis is capable of processing streaming data at in real time. This is a key challenge application developers face when they have to process huge amounts of data in real time. It can scale up or scale down based on data needs of the system. As volume of data grows with advent IOT [3] devices and sensors, Kinesis will play a key role in developing applications which require insights in real time with this growing volume of data.

Keywords: Cloud, I524

https://github.com/cloudmesh/sp17-i524/blob/master/paper2/S17-IO-3005/report.pdf

1. INTRODUCTION

Amazon Kinesis [1] helps application developers collect and analyze streaming data in real time. The streaming data can come from variety of sources like social media, sensors, mobile devices, syslogs, logs, web server logs, network data etc. Kinesis can scale on demand as application needs change. For example during peak load situation kinesis added more workers nodes and can reduce the nodes when the application runs at low load. It also provides durability, where if one of the node goes down the data is persisted on disk and get replicated when new nodes come up. Multiple applications can consume data from one or more streams for variety of use cases for example, one application computes moving average and another application counts the number of users clicks. These applications can work in parallel and independently. Kinesis provides streaming in realtime with sub-second delays between producer and consumer. Kinesis has two types of processing engines: Kinesis streams - reads data from producers and Kinesis firehose - pushes data to consumers.

Kinesis streams can be used to process incoming data from multiple sources. Kinesis firehose is used to load streaming data into AWS like Kinesis analytics, S3 [4], Redshift [5], Elasticsearch [6] etc.

2. ARCHITECTURE

2.1. Introduction

Amazon Kinesis reads data from variety of sources. The data coming into streams is in a record format. Each record is composed on a partition key, sequence number and data blob which is raw serialized byte array. The data further is partitioned into multiple shards(or workers) using the partition key.

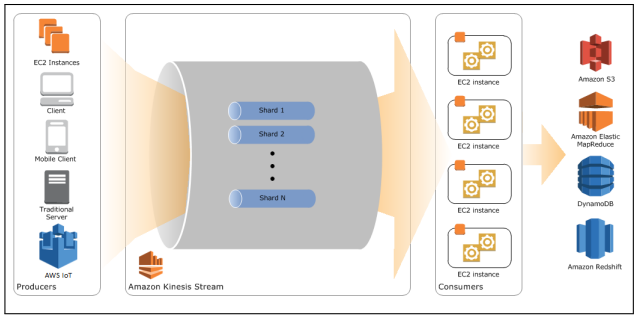

Fig. 1. Kinesis streams building blocks [7]

2.2. Building blocks

Following are key components in streams architecture [7] :

2.2.1. Data Record

Its one unit of data that flows through Kinesis stream. Data records is made up of sequence number, partition key, and blob of actual data. Size of data blob is max 1 MB. During aggregation one or more records are aggregated in to a single aggregated record. Further these aggregated records are emitted as an aggregation collection.

2.2.2. Producer

Producers write the data to Kinesis stream. Producer can be any system producing data. For example, ad server, social media stream, log server etc.

2.2.3. Consumer

Consumers subscribe to one or more streams. Consumer can be one of the applications running on AWS or hosted on EC2[8]

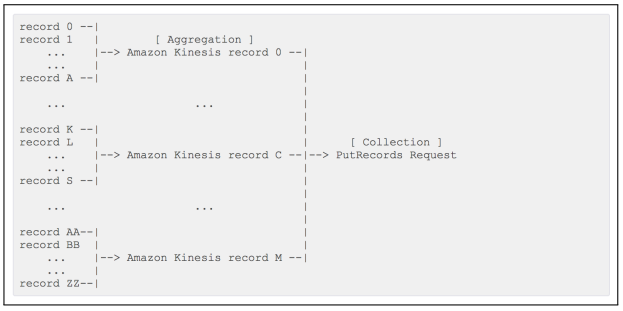

Fig. 2. Aggregation of records

instance(virtual machines).

2.2.4. Shard

A shard is an instance of kinesis stream engine. A stream can have one or more shards. Records are processed by each shard based on the partition key. Each shard can process up to 2MB/s data for reads and up to 1MB/s for writes. Total capacity of a stream is sum of capacities of its shards.

2.2.5. Partition Key

Partition key is 256 bytes long. A MD5 [9] hash function is used to map partition keys to 128 bit integer value which is further used to map to appropriate shard.

2.2.6. Sequence Number

Sequence number is assigned to a record when a record get written to the stream.

2.2.7. Amazon Kinesis Client Library

Amazon Kinesis Client Library is bundled into your application built on AWS. It makes sure that for each record there is a shard available to process that record. Client library uses dynamo db to store control data records being processed by shards.

2.2.8. Application Name

Name of application is stored in the control table in DynamoDB [10] where kinesis streams will write the data to. This name is unique.

3. KINESIS DEVELOPMENT

AWS provides a java SDK. Java SDK [11] can be used to complete all workflows on stream. Workflow like create, listing, retrieving shards from stream, deleting stream, re-sharding stream and changing data retention period. SDK provide rich documentation and developer blogs to support development on streams.

You can create and deploy Kinesis components using following: Kinesis console, Streams API, and AWS CLI.

Before creating stream you should determine initial size of the stream [12] and number of shards required to create your stream. Number of shards can be calculated using the formulae:

$$NumberOfShards = max(\frac{A}{1000}, \frac{B}{2000})$$

A = Incoming Write Bandwidth In KB
B = Outgoing Read Bandwidth In KB

Here, the attributes used in the calculation are self explanatory. Producer for streams writes data records into Kinesis streams. This data is available for 24 hours within streams. The default retention interval can be changed. To write records to stream, you must specify partition key, name of stream and data blob. Consumer on the other hand read data from streams using shard iterator. Shard iterator provides consumer a position on streams from where the consumer can start reading the data.

4. STREAM LIMITS

4.1. Shard

Kinesis streams has certain limits [13] : by default there can 25 shards in a region except US east, EU and US west have limit of 50 shards. Each shard can support up to 5 transactions per second for reads and at maximum data rate of 2 MB per second. Each shard can support 1000 records per second for writes and maximum data rate of 1MB per second.

4.2. Data retention

By default the data is available for 24 hours which can be configured up to 168 hours with 1 hour increments.

4.3. Data Blob

Maximum size of data blob is 1MB before base64 encoding [13].

5. MANAGEMENT

Kinesis provides all management using AWS console or you can build a custom management application using Java SDK [11] provided by AWS. AWS provides a web console to manage all AWS services including kinesis. Using console web user interface user can perform all operations to manage stream.

6. MONITORING

AWS provides several ways to monitor its services. Kinesis can use these services for monitoring purpose: CloudWatch metrics, Kinesis Agent, API logging, Client library, and Producer Library

CloudWatch metrics allows you can monitor the data and usage at shard level. It can collect metrics like: latency, incoming bytes, incoming records, success count etc.

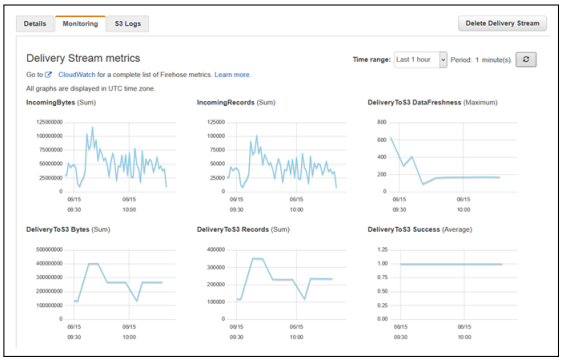

Fig. 3. Kinesis Metrics[14]

7. LICENSING

Kinesis is software as a service(SAAS) from Amazon AWS infrastructure. Hence it can only run as a service within AWS. It comes with pay-as-you-go pricing.

8. USE CASES

Kinesis streams [1] and firehose can be useful in variety of use cases: log data processing, log mining,realtime metrics, reporting realtime analytics, and complex stream processing

For example, Kinesis can be used in serving advertisements based on user click events. Where clients send the clickstream data to Kinesis streams. Stream further generates records which are processed by spark streaming. Further, the algorithms running on spark streaming can be used to generate insights based on user's interest.

> We send clickstream data containing content and audience information from 250+ digital properties to Kinesis Streams to feed our real-time content recommendations engine so we can maximize audience engagement on our sites - Hearst Corporation [1]

Kinesis solves variety of these business problems by doing a real time analysis and aggregation. This aggregated data can further be stored or available to query. Since it runs on amazon, it becomes easy for users to integrate and use other AWS components.

9. CONCLUSION

Kinesis can process huge amounts of data in realtime. Application developers can then focus on business logic. Kinesis can help build realtime dashboards, capture anomalies, generate alerts, provide recommendations which can help take business and operation decisions in real time. It can also send data to other AWS services. You can scale up or scale down as application demand increases or decreases and only pay based on your usage. Only downside of Kinesis it that it cannot run on a private or hybrid cloud, rather can only run on AWS public cloud or Amazon VPC(Virtual Private Cloud)[15]. Customers who want to use Kinesis but don't want to be on Amazon platform cannot use it.

ACKNOWLEDGEMENTS

Special thanks to Professor Gregor von Laszewski, Dimitar Nikolov and all associate instructors for all help and guidance related to latex and bibtex, scripts for building the project, quick and timely resolution to any technical issues faced. The paper is written during the course I524: Big Data and Open Source Software Projects, Spring 2017 at Indiana University Bloomington.

REFERENCES

[1] "Kinesis - real-time streaming data in the aws cloud," Web Page, accessed: 2017-01-17. [Online]. Available: https://aws.amazon.com/kinesis/

[2] "AWS - amazon web services," Web Page, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/

[3] K. Hwang, J. Dongarra, and G. C. Fox, *Distributed and cloud computing: from parallel processing to the internet of things*, T. Green and R. Day, Eds. Morgan Kaufmann, 2012. [Online]. Available: https://www.amazon.com/Distributed-Cloud-Computing-Parallel-Processing-ebook/dp/B00GNBLGE4?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=B00GNBLGE4

[4] "Amazon S3 simple durable, massively scalable object storage," Web Page, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/s3

[5] "Amazon Redshift fats, simple, cost-effective data warehousing," Web Page, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/redshift/

[6] "Amazon Elasticsearch fully managed, reliable, and scalable elasticsearch service," Web Page, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/elasticsearch-service

[7] "Amazon Kinesis streams key concepts," Web Page, accessed: 2017-03-15. [Online]. Available: http://docs.aws.amazon.com/streams/latest/dev/key-concepts.html

[8] "Amazon EC2 secure and resizable compute capacity in cloud," Web Page, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/ec2/

[9] R. Rivest, "Rfc 1321," *The MD-5 Message Digest Algorithm, SRI Network Information Center*, no. 1321, Apr. 1992. [Online]. Available: https://tools.ietf.org/html/rfc1321

[10] "Amazon S3 fast and flexible NoSQL database," Web Page, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/dynamodb/

[11] "Kinesis - aws sdk for java," Web Page, accessed: 2017-03-15. [Online]. Available: https://aws.amazon.com/sdk-for-java/

[12] "Kinesis - real-time streaming data in the aws cloud," Web Page, accessed: 2017-03-15. [Online]. Available: http://docs.aws.amazon.com/streams/latest/dev/amazon-kinesis-streams.html

[13] "Amazon Kinesis streams limits," Web Page, accessed: 2017-04-01. [Online]. Available: http://docs.aws.amazon.com/streams/latest/dev/service-sizes-and-limits.html

[14] B. Liston, "Serverless cross account stream replication using aws lambda, amazon dynamodb, and amazon kinesis firehose," Web Page, Aug. 2016, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/blogs/compute/tag/amazon-kinesis/

[15] "Amazon virtual private cloud (VPC)," Web Page, accessed: 2017-04-01. [Online]. Available: https://aws.amazon.com/vpc/