Leveraging Cloud-Based Predictive Analytics to Strengthen Audience Engagement

By U. Shakeel and M. Limcaco

Abstract

To grow their business and increase their audience, content distributors must understand the viewing habits and interests of content consumers. This typically requires solving tough computational problems, such as rapidly processing vast amounts of raw data from Web sites, social media, devices, catalogs, and back-channel sources. Fortunately, today's content distributors can take advantage of the scalability, cost effectiveness, and pay-as-you-go model of the cloud to address these challenges. In this paper, we show content distributors how to use cloud technologies to build predictive analytic solutions. We examine architectural patterns for optimizing media delivery, and we discuss how to assess the overall consumer experience based on representative data sources. Finally, we present concrete implementations of cloud-based machine learning services and show how to use the services to profile audience demand, to cue content recommendations, and to prioritize the delivery of related media.

This raises the question of how to build the next-generation media delivery platform that not only delivers reliable content but also ensures that the content is experienced the way the content creator intended.

Introduction

An abundance of technical advancements has expanded the range of options for media consumers. Today's consumers can choose to have 3-D, 4K, HDR, or even 8K content displayed on a variety of sophisticated devices. Given the public's appetite for these high-end devices, media creators are constantly under pressure to increase resolution and quality to compete in an ever-expanding war of content choices.

In addition to the changes in display technologies, on-demand content and streaming media delivery have changed the habits of content viewers. Gone are the days when we used to circle around the TV set at an appointed time for the airing of our favorite TV show. People now expect to watch the programming they want on their own schedule, which is driving more and more media companies to consider providing their own over-the-top service. These services use the internet for delivery, which introduces potential quality issues that are out of the control of the media owner or distributer. To mitigate these risks, many media distributors invest heavily in solutions that detect playback issues; these solutions require large amounts of computational capacity to process massive, raw datasets to provide real-time course correction. In this way, distributors can provide more reliable content that caters to the viewing habits of their audience.

This raises the question of how to build the next-generation media delivery platform that not only delivers reliable content but also ensures that the content is experienced the way the content creator intended. One approach is to have the delivery platform predict when events, such as network congestion or low-quality streams, will occur in the future, and subsequently guide consumers in the right direction. Powerful tools toward this goal include using data generated by consumers from their interaction with content, social media, and multiple screens in conjunction with predictive modeling, machine learning, and real-time analytics. According to Nielsen, social media activity drives higher broadcast TV ratings for 48% of shows;[1] in a similar survey by Netflix, over 75% of what people watch is based on Netflix's recommendations.[2] In terms of audience engagement, there are two basic categories:

- *Content experience*—Using predictions and analytics on viewers' viewing habits, player network logs, and datasets to quickly analyze existing issues or predict future issues. The predictions can be used to minimize or even eliminate a poor customer experience.

Digital Object Identifier 10.5594/JMI.2016.2602121
Date of publication: XXXX

- *Content relevance*—Using predictions and analytics on historical and some real-time datasets to detect and recommend relevant content and personalize content and ads, thereby improving the experience for content selection.

Audience Engagement Signals

To build a next-generation media delivery platform and deliver a better customer experience, content distributors can capture, fuse, and synthesize background signals (noise) to create models to analyze, for both batch and real-time data. We can leverage both transactional data (from user interactions such as searching, playing, watching/listening, and contacting sales/support) to behavioral activities (such as sharing, tagging, liking, reviewing the content, and so on) to build a prediction model. The analysis can be descriptive (aggregation, retrospective), predictive (statistical, machine learning), or prescriptive (what should we do about it) based on technology choices. In the remainder of this paper, we focus on descriptive and predictive analytics, especially in the context of content relevance.

Machine Learning for Predictive Analytics

Machine learning (ML) is a broad area of tools and techniques that can help you use historical data to make better business decisions. ML algorithms discover patterns in data and construct predictive models using these patterns, allowing you to use the models to make predictions from future data. For example, you could use ML to predict whether customers will select a title to view based on data such as their viewing history, what other users in their same demographic have watched, and even whom they follow on social media platforms. You then can use the predictions to identify which customers are most likely to respond to your personalized, promotional e-mail.

Benefits of the AWS Cloud for Predictive Analytics

Amazon Web Services (AWS) offers a comprehensive, end-to-end portfolio of cloud computing services that you can use to analyze your consumers' interactions. AWS performs high-volume data processing at scale and at a fraction of the cost compared with traditional data analytics infrastructure solutions. It is important to understand the scale of such a dataset. For example, as of 2011, Netflix has been managing 20 billion requests per month for millions of consumers across more than 60 geographies.[3] Netflix leverages tens of thousands of EC2 virtual instances on demand and terminating the instances when the work is complete.

Analyzing large datasets requires significant compute capacity that can vary in size based on the amount of input data and the analysis required. This characteristic of big data workloads is ideally suited to the pay-as-you-go cloud-computing model of AWS, where applications can easily scale up and down based on demand. This elasticity means that as requirements change, you can easily resize your environment (horizontally or vertically) on AWS to meet your needs without having to wait for additional hardware or overinvest to provision for peak capacity. This scalability is especially important for mission-critical applications. In contrast, system designers for traditional infrastructures have no choice but to overprovision because systems must be able to handle surges in data volumes due to increases in business demand.

In addition, the dynamic computing resources of AWS run on a world-class infrastructure, which is available across the 11 different geographic regions that AWS supports. The AWS platform offers several scalable services, such as Amazon Simple Storage Service (Amazon S3) and AWS Data Pipeline. The scalability, elasticity, and global footprint of the AWS platform make it an extremely good fit for solving big data problems. You can read about how many AWS customers have implemented successful big data analytics workloads on the AWS Case Studies & Customer Success Stories, Powered by the AWS Cloud Web page. AQ 1

Technology

Relevant Products and Solutions

The technology spectrum in this space can be categorized into three main areas:

- Desktop-driven data science tooling (including Microsoft Excel, KNIME, IBM SPSS Statistics, RapidMiner, R language and environment, Weka, MATLAB, and Octave)
- Server-side big data platforms (several products within the Hadoop platform, such as Apache Mahout, Spark MLlib, Oxdata, GraphLab, R+ Hadoop, Radoop, Apache Hama, Apache Giraph, Apache HBase Kiji, and BigML)
- Scoring/prediction deployment services (Zementis ADAPA, and Orynx).

Many of these options (Prediction IO, Mortar, BigML) are available in the AWS Marketplace as well. AWS Marketplace allows users to launch Amazon EC2

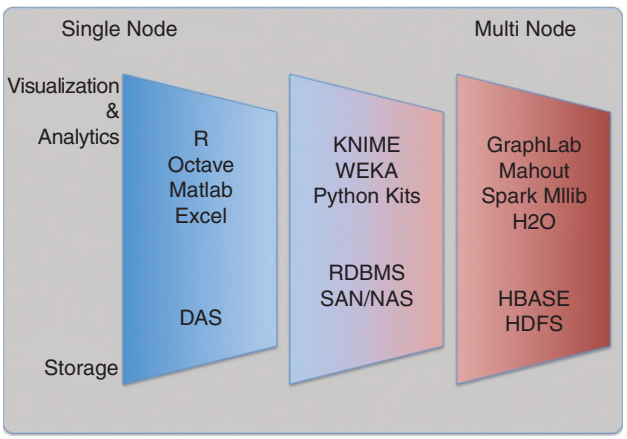

FIGURE 1. Relevant analytics technologies.

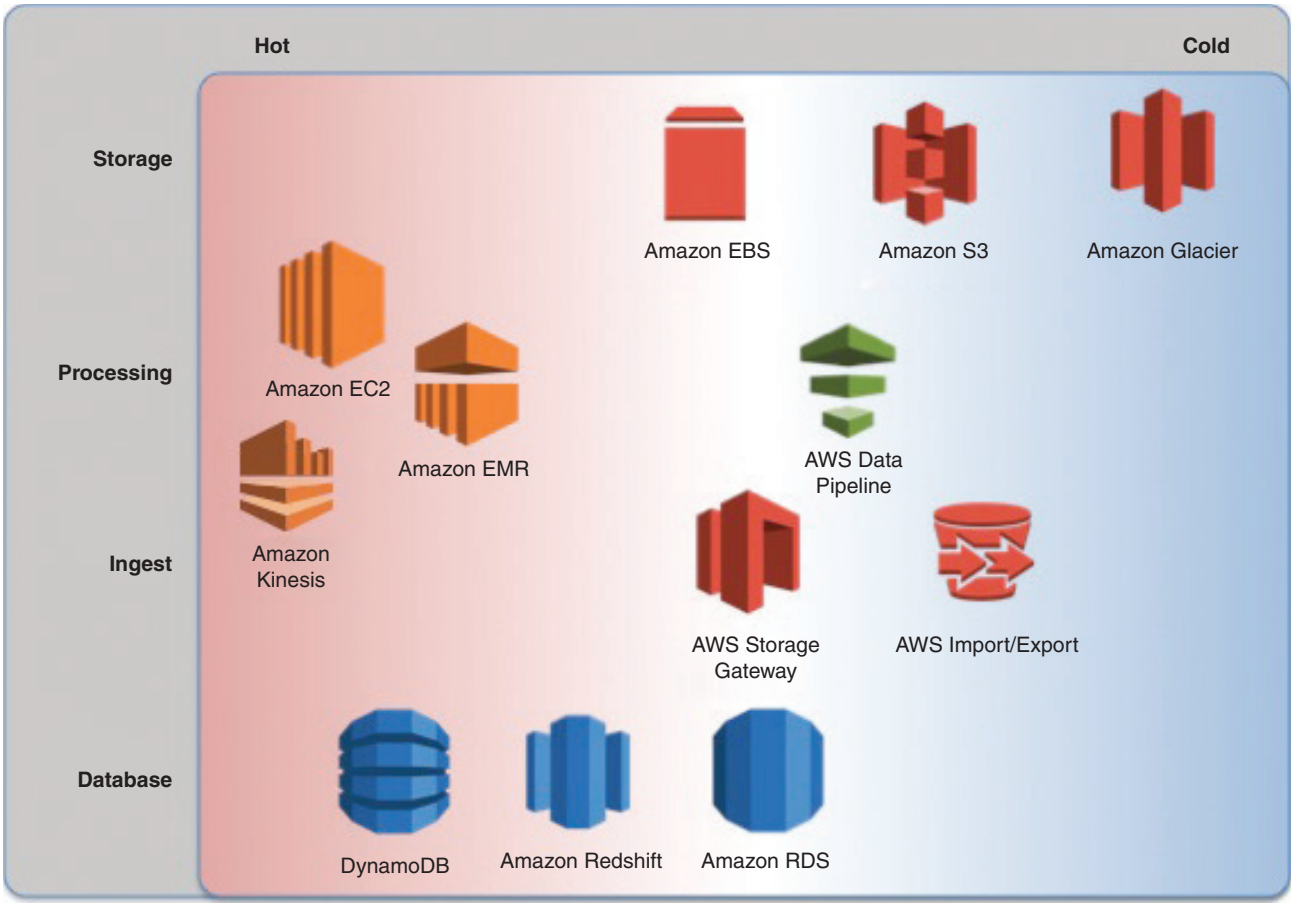

FIGURE 2. Relevant services: AWS.

instances or Amazon EMR workloads with software licensing included in the hourly pricing model, so you simply pay a prorated license fee for the time you use the application. You can choose appropriate products specific to the use case, depending on whether you need a single node job or a distributed, clustered job.

Relevant Services on the AWS Platform

AWS data analytics solutions consist of fully managed services, including scalable data ingest, storage, RDBMS, data warehousing, NoSQL, analytics, and archival. All these individual solutions leverage the powerful scale and utility services of AWS. There are several important AWS services for building big data solutions, as described in the following sections.

Amazon Kinesis[4] is a managed service for real-time processing of streaming big data. This service supports data throughput from megabytes to gigabytes of data per second and can scale seamlessly to handle streams from hundreds of thousands of different sources. Kinesis provides high availability and durability data streams in a cost-effective manner.

Amazon Simple Storage Service (Amazon S3)[5] is the ideal big data cloud storage solution to store original content durably. Designed for eleven 9's of durability, with no single point of failure, S3 is a fundamental big data object store as well as an ideal storage solution for digital content (source and processed content files).

Amazon DynamoDB[6] is a fast, fully managed NoSQL database service that makes it simple and cost-effective to store and retrieve any amount of data or serve any level of request traffic. DynamoDB has provisioned guaranteed throughput and single-digit millisecond latency, which makes it a great fit for gaming, ad tech, mobile, and many other big data applications. All data items are stored on solid-state disks.

Amazon Redshift[7] provides a fast, fully managed, petabyte-scale data warehouse for less than $1,000 per terabyte per year. Redshift delivers fast query and I/O performance for virtually any size dataset by using columnar storage technology and parallelizing and distributing queries across multiple nodes.

Amazon Elastic MapReduce (Amazon EMR)[8] provides the powerful Apache Hadoop framework on EC2 as an easy-to-use managed service. With EMR, you can focus on your map/reduce queries and take advantage of the broad ecosystem of Hadoop tools, while deploying to a high-scale, secure infrastructure platform. The Amazon Spot Market,[9] which is integrated into EMR, lets you choose your own price for the computing resources you need to do analytics with cloud computing.

Amazon Machine Learning (Amazon ML) is a service that makes it easy for developers of all skill levels to use ML technology. ML provides visualization tools and wizards that guide you through the process of creating models without having to learn complex ML algorithms and technology. The service is based on the same proven, highly scalable ML technology used for years by the data scientist community at Amazon. The service uses powerful algorithms to create ML models by finding patterns in your existing data. ML uses these models to process new data and generate predictions for your application.

Deep Dive: Use Case Solutions

In this section, we focus on the use case for content relevance, and we examine AWS features and their value proposition in the light of architectural design patterns.

Content Relevance

Let us start by looking at how you can generate predictions based on an analysis of historical and real-time datasets to detect and recommend relevant content as well as to personalize content.

Data sources

One of the key questions is "What data is meaningful for determining content relevance?" The answer typically is data about user likes/dislikes, user interactions on social media channels, and users' viewing habits correlated to user profiling (because they like *x*, they may like *y*, and so on). However, this can result in a very large dataset, both historical as well as real-time, when you gather data across all consumers of your content. AWS provides services, such as Kinesis and the highly scalable fleet of EC2 instances, that give you the means to ingest very large quantities of data and scale on demand in a highly available and durable fashion. Additionally, AWS provides scalable connectivity to its storage services like Amazon S3 and Amazon Glacier, which you can leverage for historical data. You can also use these services to transition real-time data to archives to make the data available for historical analysis.

AQ 2

Social media

Today's gadget-savvy customers are connected to their peers, family, and friends seemingly all the time, even while they are consuming content. Social media sentiment is an important source of audience interest. Social media channels like Facebook, Twitter, Instagram, Pinterest, Foursquare, Tumblr, Google Plus, IMDB, and Flicker are common examples. Most of these social media channels provide API access (RESTful, in some cases) that you can use to get data about near real-time user behavior. You can then filter the resulting dataset, to gather signatures for a specific piece of content.

Media player logs

Most video players provide the capability to capture and stream a real-time dataset to a backend ingestion point. The dataset can contain anything from seek, play, pause, and other player controls to even the specific bitrate delivered to the customers (in the case of adaptive bitrate). Most off-the-shelf media players provide this capability; however, many distributors also build their custom players to include additional data collection and reporting capabilities for interactive experiences like search, on-the-fly recommendations, camera angle selection, or even shopping experiences, effectively creating an interactive knowledge dump about the content. Data describing how users interface with these controls and engagement points can be streamed back to the backend analytics engines and environments.

Viewing history

Viewing history can be captured by the front-end application or the medium that the viewer uses to navigate and select the content. You can build recommendations based on this historical data and the historical capture of previous recommendations resulting into views.

Approaches

Once the data has been captured (both historically or streamed in realtime), the challenge is to make sense of the raw data (from the data sources discussed above). Depending on the turnaround time required for the underlying application, AWS scalability can be of great advantage to spin up clusters of hundreds of compute nodes running the analytics engine of your choice in a matter of minutes.

Sentiment analysis

From a technical standpoint, the concept behind sentiment analysis is to process largely unstructured natural language sources and to extract subjective meaning behind the words being expressed. It is often used as a means of automatically gauging user interest and community trending on topics related to events, products, and personalities in the social media space. You can then apply this understanding in near real-time toward use cases ranging from content recommendations to digital media ad campaign efficacy.

Analysis in this context involves the application of text processing and ML techniques to classify audience commentary by using a generalized understanding of what is deemed positive versus negative sentiment. Natural language processing itself is a challenging subject, with numerous studies dedicated to obtaining high precision in the extraction and understanding of meaning behind the spoken and written word. This is especially complicated in the social media space with format-limiting colloquial expressions (tweets). The other challenge comes from the large volume of this content, often streaming from a variety of outlets: tweets, posts, blogs, and movie reviews.

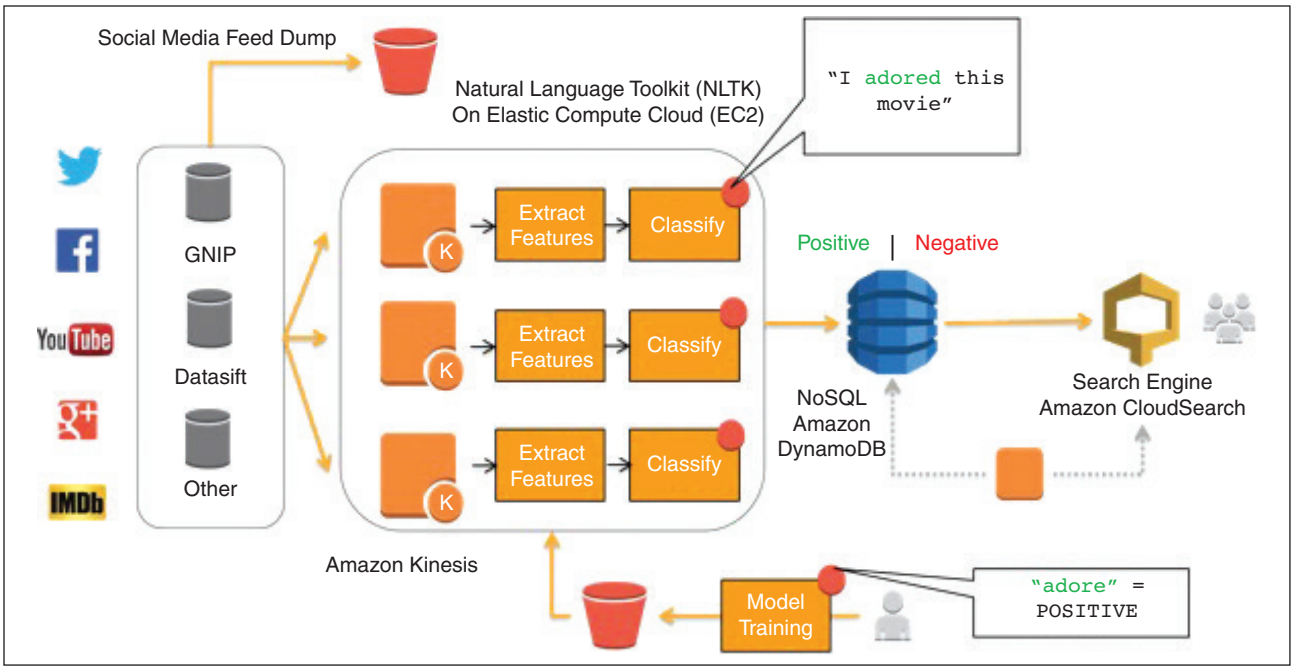

FIGURE 3. Sentiment analysis on real-time sources.

One general technique to deal with this is the application of conditional probabilities as a "bag of words" concept. That is, a system is initially trained with examples of both positive and negative sentiment, and these text examples are tokenized and treated as simple collections of words. The system uses this baseline collection of words and word frequencies to determine whether new word collections are likely examples of either category. You can use numerous programming kits and libraries that implement the "Naïve Bayes" classifier to this effect, such as the Python-based Natural Language Toolkit;[11] however, the goal of performing this operation in near-real-time at high volumes remains a challenge. The proposed approach to this is to perform the learning process offline with a smaller subset of the overall corpus and then to apply the trained model inline in a high-volume stream ingest processing pipeline.

Training the system with different types of sentiment is performed out-of-band from a few thousand to ideally several hundred thousand samples. Several predefined training sets are generally available from generic and academic repositories; however, to train using very specific industries, it is desirable to use more specific sources. You may, for example, elect to crowdsource the process of inspecting a phrase and classifying the intent. Amazon Mechanical Turk is a cloud-based human workflow system that allows you to specify tasks up for bid, which then can be executed by a vast global workforce connected through this service. You can programmatically allocate specific classification tasks with required parameters, process the community feedback, and automate the creation of the training model data structure using this information. Once your initial model is created (assessed and fine-tuned), you then can deploy this into a real-time flow. Data sources, such as Twitter and Facebook, and managed data brokers, such as Datasift, provide interfaces for ingesting large volumes of social media commentary. You can ingest this data and filter it for specific attributes, phrases, and sources. The key is to receive and process the data and apply your trained model on the inbound stream at high velocity. AWS provides a managed ingest messaging service to collect and process large streams of data records in realtime.

In this logical architecture, you receive data through established APIs from social media sources. You then apply the pretrained Naïve Bayes Classifier model against the continuously updated stream of sentiment. The training logic is embedded in simple callback handlers registered with the Kinesis streaming service by using the Kinesis Client Library framework. Classifications (positive or negative) are updated in near realtime against a high-throughput data store such as a NoSQL system like DynamoDB. You can use the data in several ways: for example, you can make timely queries to inform digital marketing strategists on the efficacy of media efforts to date, and you can filter or affect content/video recommendations as part of an over-the-top (OTT) experience. (We discuss recommendations in greater detail in a later section.)

As an alternative to this Natural Language Toolkit (NLTK) Naïve Bayes implementation, you may also employ a more composite service to achieve these goals. Amazon ML supports the ingestion of text sources and can be trained by using industry-standard logistic regression algorithms to characterize sample unstructured streams of sentiment. Amazon ML provides the tools for processing

source CSVs containing positive and negative examples, and evaluates and automatically generates a model for use during realtime or batch prediction operations.

Segment audience

It is often desirable to automatically categorize subgroups contained within a larger audience population to more readily engage these members with personalized content and targeted initiatives (i.e., digital marketing campaigns). ML provides several approaches. You can, for example, use a form of unsupervised learning known as "*k*-means clustering" to automatically and recursively identify audience member affinities across an *n*-dimensional space. *K*-means clustering seeks to categorize entities based on a defined similarity measure; the "*k*" represents the seeded number of target groups that the system then iterates on and converges until *k*-number of segments are discovered. However, when the size of the overall population and the breadth of user attributes (geographies, demographics, viewing behavior, and so on) is large, this can exceed the capabilities of a desktop toolset such as R or Weka. Here, you can burst into the cloud and extend the reach of the client toolset by delegating the processing to a scalable backend ML cluster readily capable of processing millions of records. One such example of this configuration is in the combined R + 0xdata H_20 platform. 0xdata provides fast, distributed scale-out k-means clustering on EC2 and can seamlessly present results back to an analyst using the R desktop client for visualization and final disposition.

In addition to leveraging previously learned audience segment characteristics, a good recommendation system leverages other audience signals on an ongoing basis to optimize and personalize the user experience.

Applications

Based on these approaches, let us analyze how you can use this normalized data across audience segments to deliver some real-world applications in the content delivery space.

Recommendations and interactive experiences

Content recommendations are an important part of overall audience experience. Building this as a part of any media platform provides a great opportunity to optimize the overall user experience and to strengthen overall engagement. In addition to leveraging previously learned audience segment characteristics, a good recommendation system leverages other audience signals on an ongoing basis to optimize and personalize the user experience. Such signals include explicit user interactions (the user purchases a movie) as well as implicit user interactions (the user searches for a genre). You can collect and analyze these signals on an ongoing basis by generating history and observation matrices, and then you can find similar items and users based on these observations to recommend new items of interest. Of course, the growing volume of subscribers and content in current and emergent OTT/broadcast scenarios drives us

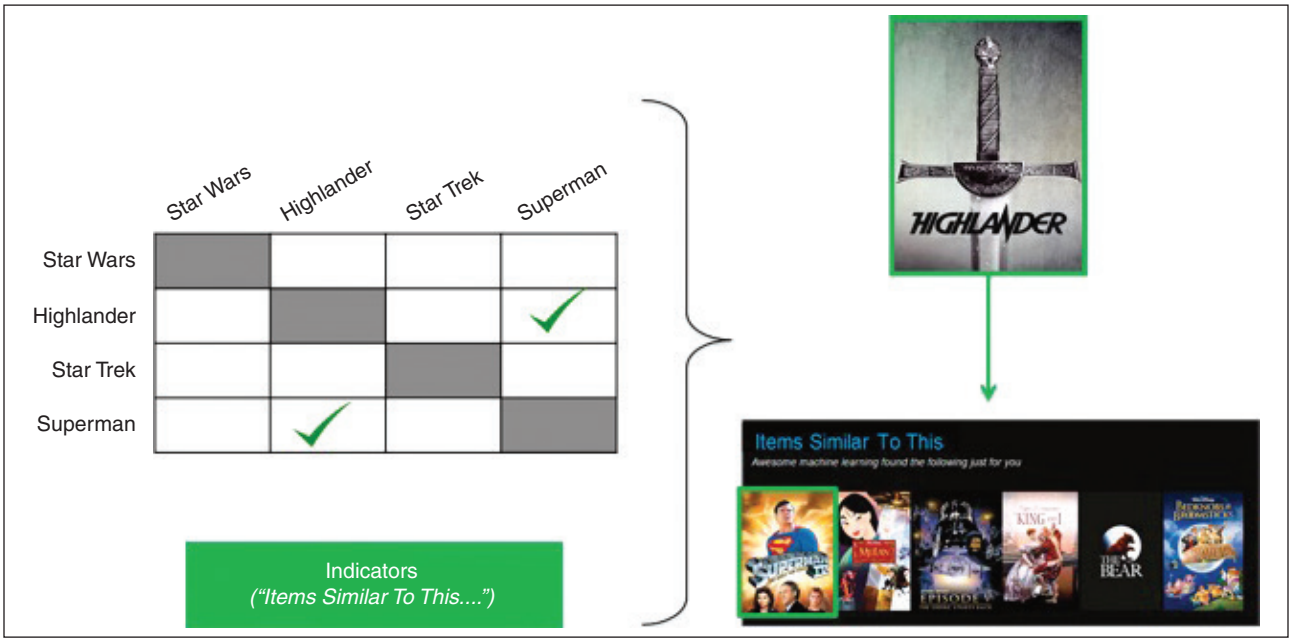

FIGURE 4. Finding interesting pairs of items ("co-occurrences").

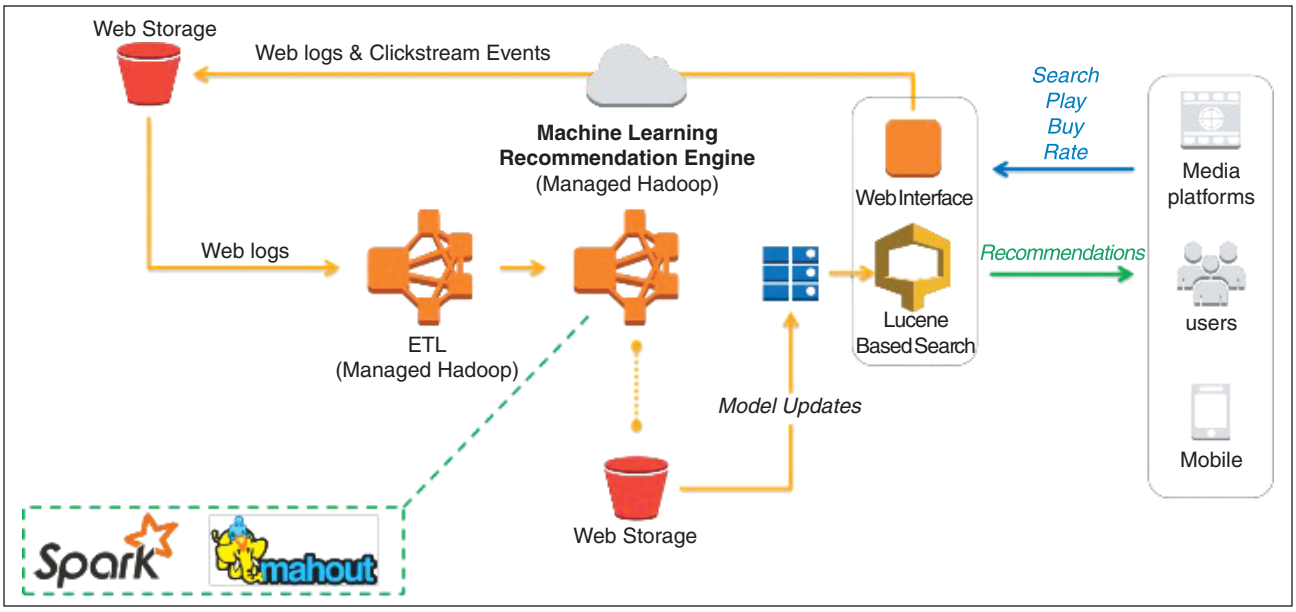

FIGURE 5. Integrated recommendation architecture.

to consider techniques designed to scale to the large quantity of engagement signals.

One technique uses a distributed cluster of compute nodes to process high-volume log file inputs (sourced from players and Web service infrastructure) to create "other people also liked these things" types of recommendations.[12] In this example, we use the open source, distributed ML project Apache Mahout to transform historical observations (user-A watched video-stream-A) to create a co-occurrence matrix that describes what items have been observed with others (video-stream-A co-occurred with video-stream-B in user-B's experience). We further refine this matrix to surface the most interesting co-occurrences using the log-likelihood ratio (LLR) measure. This effectively gives us the baseline data structure to calculate item similarity results—for example, the "items similar to this" in a typical online media or e-Commerce experience. Specifically, we can utilize Apache Mahout's *spark-itemsimilarity* function to process source log data stored in a Web store like Amazon S3. Using a backend Hadoop cluster (Amazon EMR) to host the infrastructure, you can process large amounts of historical logs, cleanse and normalize the data, and then pass it into the ML pipeline. The Mahout process converts the log data and presents output results in the form of *<item> <appeared with item-1> <item-2> <item-3>* and so on.

When coupled with a search engine and a record of historical user interactions, you also can use this data to deliver personalized results. That is, by tracking user selection history (clicks, downloads, wish lists, plays) you can generate a growing list of implicit search criteria that can be supplied to a backend search engine capable of finding similar items of interest. In this case, the backend search index is supplied with data from our item similarity data structure from the offline process that we described earlier. You can load this data structure into an online search engine, such as Solr, Elasticsearch, or a managed search service, such as Amazon CloudSearch. By applying concepts such as term frequency—such as inverse document frequency (tf-idf)—you can employ an implicit search against your index. The search provides a form of distance measures and answers the following question: "Given the input set of movies I'm interested in, what other movies have been observed in interesting quantities that match?" Served out of a search engine, the result is a dynamic user experience that more or less reflects recent and ongoing user interaction by providing lists of targeted content and then adjusting accordingly.

Personalized ads

You also can use the preceding approaches of user sentiment analysis and segmentation to deliver personalized ads. For example, a player requests an ad based on the user profile signature, and the ad-server returns a personalized ad in the context of a playback experience to a live scenario, where the ad-markers can be replaced with a personalized ad on the fly. Media monetization

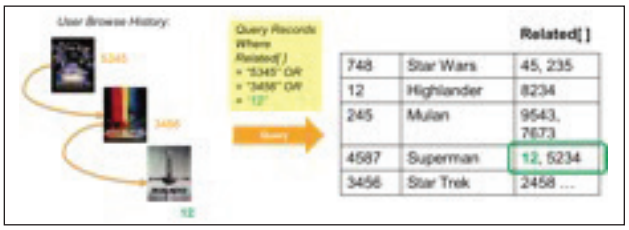

FIGURE 6. Using user online clicks to search an item similarity matrix. AQ 3

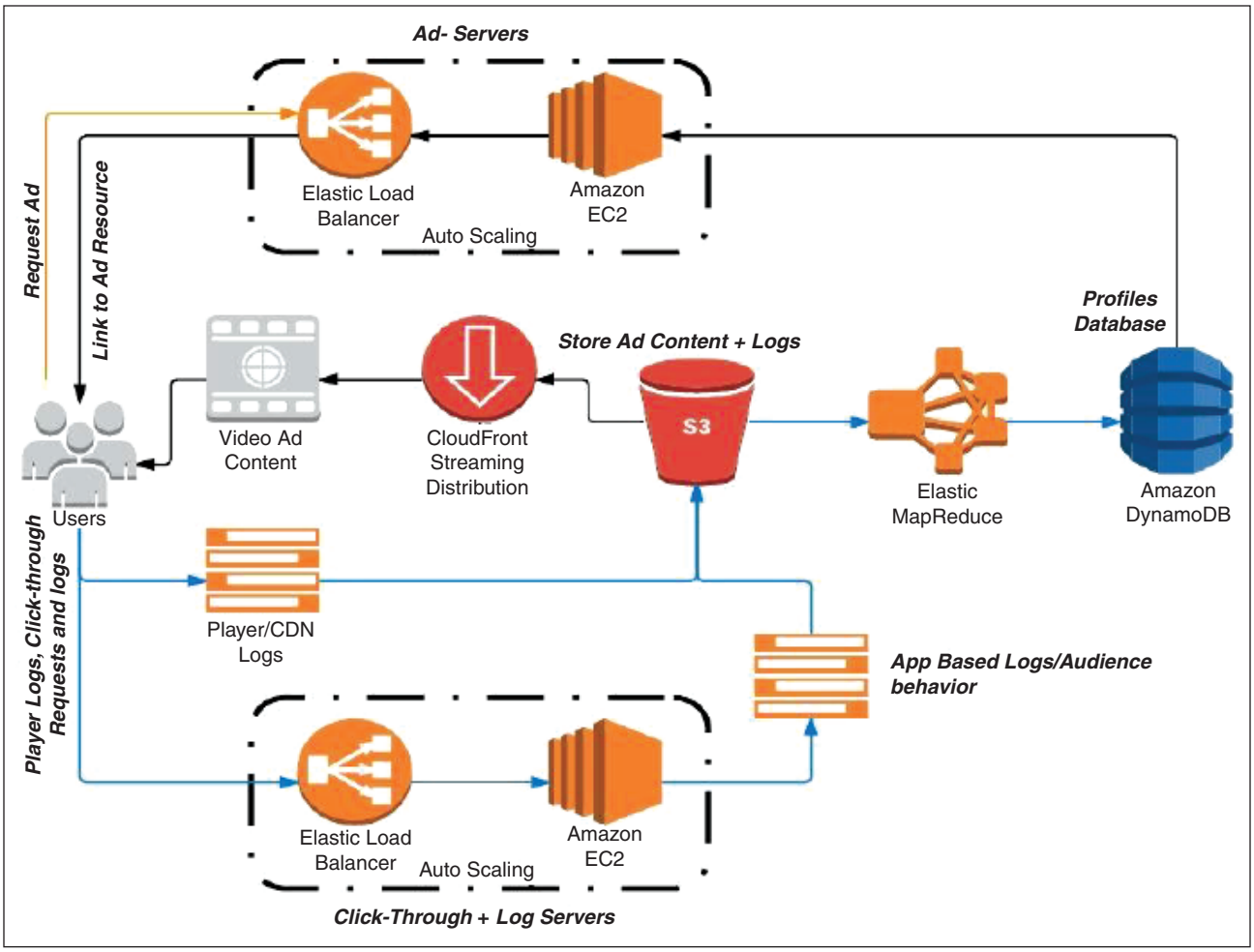

FIGURE 7. Reference architecture for ad-serving on AWS.

is often ad-driven, and a personalized, targeted ad has better chances of eventual conversion. Although the content player can enforce the serving of ads before the actual content starts playing or in between during the ad-breaks/ad-markers in the stream, it does not always mean that the audience is engaged with the content of the ads. Advertising companies spend a lot of money coming up with creative, catchy ways of attracting their audiences, but more than that, the ad content has to appeal to, and be consumed by, the right audience. You can use content recommendations across the player logs to mine ad content behavior and create signatures based on different demographics. These and other ad tech-related processing demand requires expansive compute and storage resources. Video ad platforms, such as Brightroll, achieve this kind of scale by using EMR and S3 to manage over 25 billion video ad inventory requests per month.[13] Figure 7 shows a reference architecture for dynamic ad-serving.

Content Experience

When you use predictions and analytics on datasets from users' viewing habits and player or network logs, you can quickly analyze existing issues or predict future issues. You then can take actions to minimize or even eliminate a poor customer experience. Distributors like Netflix use content delivery network (CDN) logs, media player logs, and bounce-rate data to build a model of churn detection and network performance for a particular geography and audience segment. The analysis can be done in realtime, which leads to realtime actions around CDN switching to provide a better customer experience (for example, quality in the case of ABR, or less buffering). Additionally, based on the user viewing habits or recommendations across a segment of users, the content can be pushed to the appropriate CDN pops to provide a better streaming experience. In the race to engage customers with your content, you can win or lose a customer based on the time it takes a player to start streaming your content after a user selects it. If it takes too long, you risk losing the user to a competitive channel.

Conclusion

ML tools and techniques can help you strengthen your audience engagement. The ever-expanding volume,

variety, and veracity of audience signal data, however, forces you to re-evaluate and expand your current approaches. Readily available compute and storage resources in the cloud allow these ML technologies to perform with unprecedented scale and, in many cases, improved accuracy. Tooling, such as Apache Mahout, SparkMLlib, and Oxdata H_20, provide a broad spectrum of ML techniques (such as k-means clustering, collaborative filtering-based recommendations, and logistic regression), but gain true scale and throughput improvements when deployed on cloud platforms like AWS. Moreover, cloud-native services, such as Amazon EMR and Amazon ML, provide even greater ease and convenience for data science and digital marketing specialists, freeing them from the drudgery of managing infrastructure. Instead, they can focus on discovering new audience insights and delivering solutions that strengthen overall engagement.

References

1. J. Koetsier, "Nielsen: Tweets drive higher broadcast TV ratings for 48% of shows," Aug. 6, 2013, http://venturebeat.com/2013/08/06/nielsen-tweets-drive-higher-broadcast-tv-ratings-for-48-of-shows/
2. C. Albanesius, "75 Percent of Netflix Viewing Based on Recommendations," Apr. 9, 2012, http://www.pcmag.com/article2/0,2817,2402739,00.asp
3. U. Shakeel and S. Redhakrishnan, "Maximizing Audience Engagement in Media Delivery," Nov. 14, 2013, http://www.slideshare.net/AmazonWebServices/maximizing-audience-engagement-in-media-delivery-med303-aws-reinvent-2013-28622676
4. Amazon Kinesis, http://aws.amazon.com/kinesis/
5. Amazon S3, http://aws.amazon.com/s3/
6. Amazon DynamoDB, http://aws.amazon.com/dynamodb/
7. Amazon Redshift, http://aws.amazon.com/redshift/
8. Amazon EMR, http://aws.amazon.com/emr/
9. Amazon EC2 Spot Instances, http://aws.amazon.com/ec2/purchasing-options/spot-instances/
10. Amazon Glacier, http://aws.amazon.com/glacier/
11. Source code for nltk.classify.naivebayes, http://www.nltk.org/_modules/nltk/classify/naivebayes.html
12. "Intro to Cooccurrence Recommenders with Spark," https://mahout.apache.org/users/algorithms/intro-cooccurrence-spark.html
13. AWS Case Study: BrightRoll, http://aws.amazon.com/solutions/case-studies/brightroll/

About the Authors

Usman Shakeel is a global technical leader, Media and Entertainment (M&E), with Amazon Web Services (AWS). In his current role, Shakeel is responsible for building AWS's technical strategy for M&E solutions. He also assists media customers globally by providing architectural guidance and design recommendations for leveraging cloud-based services, tools, and techniques. Shakeel has been with AWS for more than six years and has been involved in several large-scale projects involving lift and shift as well as new implementations of media workflows on AWS for some of the major Hollywood Studios. Shakeel has a master's degree in computational geometry and more than 20 years of IT experience working with various organizations in software development and research and development around computer visualization as well as software and cloud architecture.

Mike Limcaco is a senior manager, Enterprise Solutions Architecture with Amazon Web Services. In his current role, Limcaco assists media customers in the Los Angeles area by providing architectural guidance and design recommendations for leveraging cloud-based services, tools, and techniques. Current clients include members of the M&E, broadcast and interactive media industry. Limcaco has more than 25 years providing large-scale systems and software engineering expertise for public sector, commercial, and start-up organizations through various leadership/consulting roles with General Dynamics, Deloitte & Touche, Ernst & Young, and IBM.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ 1: Please confirm whether the referenced Web page is correct; otherwise, please provide the landing page that you want readers to refer to. (There are several case studies categorized by customer and industry.)

AQ 2: Please advise whether Reference 10 is cited at the end of the sentence that begins as, "
Additionally, AWS provides scalable connectivity to its storage services .." Currently, no in-text citation exists for Reference 10.

AQ 3: Readers might find the image for Figure 6 illegible. Please submit a clearer image to improve readability.