International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5S, May 2019

# Device Contextual Content Publishing in Media & Publishing Industry using Big Data Analytics on AWS

**Girish G, M. Prabhakar**

***Abstract:*** ***Media & Publishing industry was traditionally a Paper and Print Industry. Since the revolution of Internet, industry started moving print to the digital form. Ever since the rapid penetration of mobile phones the media industry has rapidly scaled down paper publishing and adopted digital form successfully***

***Internet speeds have also increased the adoption of Digital Print's. With Newspapers being accessed globally in its digital form, it is extremely important for publishers to keep their content readily accessible and rich for various devices – Tablets, Laptops, Desktop's, Mobile Phones, Smart Watches, Digital reader's etc.***

***This Paper talks about an ECONOMICAL & HIGHLY SCALABLE Big Data analytics implementation using AWS Elastic Map Reduce (EMR) to derive trends on end user usage patterns and choice of device. This will help the publishers rapidly scale to provide device contextual content to end users with ever changing access mechanisms***

***Indexed Terms— AWS-EMR, BigData, Device-Contextual, Media&Publishing***

## I. INTRODUCTION

With the advancement in Internet technologies and available bandwidth worldwide, the biggest beneficiaries have been the Media and Publishing (MP) Industry. This industry has seen a huge change in its subscribers accessing its content. While traditionally this industry relied on paper and print mediums, today this medium is fast diminishing and being replaced by electronic medium which is exponentially growing both in size and variety

The MP industry will need to cater to this change and have the ability to meet the pace of change. This is where Big Data Analytics has a very huge role to play. Effectively leveraging Big Data analytics, publishers can understand the different mediums and variations used to access the context and customize their content to match the mediums.

It is not economically feasible to build an On-Premise Big Data solution which is scalable to meet the changing user base and multitude of usage patterns and devices.

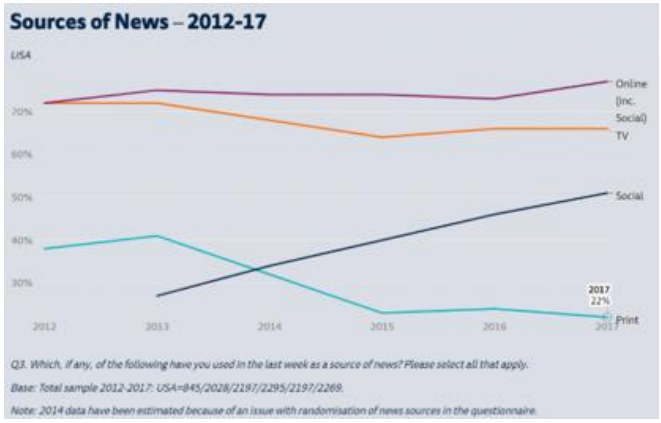
This paper focusses on how the MP industry can leverage the AWS- EMR offering for faster setup and scale on demand while at the same time keep the costs low for their analytics submission.

**Manuscript published on 30 May 2019.**
***** Correspondence Author (s)
**Girish G,** School of C& IT, Reva University, Bangalore, India.
**Dr. Prabhakar M,** School of C&IT, Reva University, Bangalore, India.



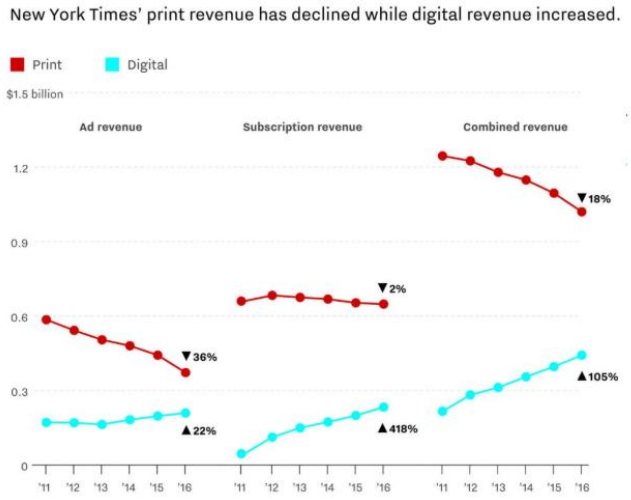
## II. MARKET TRENDS DRIVING CHANGE IN MEDIA AN PUBLISHING INDUSTRY

*Figure 2.1 Media and Publishing Evolution*




*Source: Reuters Instititue digital News Report - 2017*

*Figure 2.2 Impact of Online-Digital for a large US based news company – New York Times*

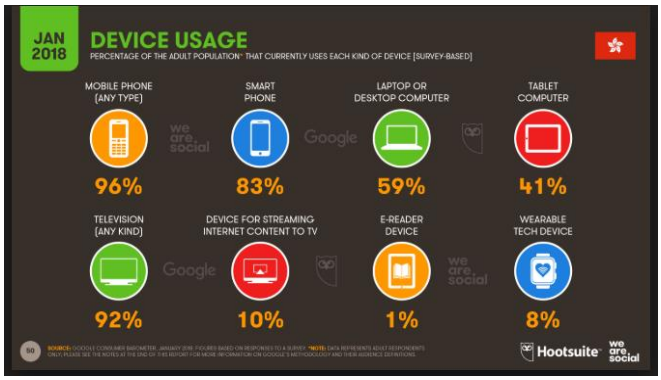



*Source: New York Times Company*

As you can see from the reports above, there is a

- 22% drop in print consumption from 2014 to 2017
- Online consumption has increased to 90%
- Print revenue for New York times has reduced by 18% while online revenue has increased by 105% between 2011-16

*Figure 2.3 Device Usage Statistics*




*Source: Google Consumer Barometer – January 2018*

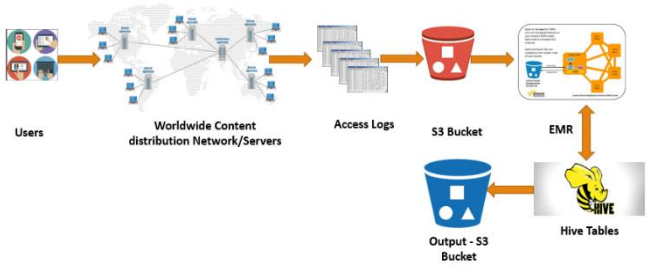
Following are some of the key highlights of the Google consumer barometer report

- 96% of adults use Smart Phones and 83% of them use smart phones
- 8% of adults use wearable tech
- 59% use Desktop/Laptops

Another important thing to note in addition to various device usage, which makes Big Data even more important is the number of Operating System Variants and Emergence of new Operating systems especially for wearables which are touted to be the future of content access medium.
Ex: There are about 185 versions of various operating systems and their versions put together only for mobiles

## III. HIGH LEVEL ARCHITECTURE FOR IMPLIMENTATION

*Figure 3.1 High level Architecture*




The high level architecture for Implementing an economical Big Data analytics solution on AWS is as shown above

**The Users** depicted in the architecture diagram above are end users who are accessing the content from various device/s from multiple locations. At any point of time there could be more than thousands of concurrent sessions

**Content distribution Network:** Content distribution Networks or CDN are typically edge servers on which the Media and Publishing companies store their data.
As the content which are Published are rich and contain high resolution pictures, high definition videos, it is practically not feasible to store the content on a centralized server and push the content when requested. This will need huge bandwidth connections and also will introduce latency which will hamper end user experience. This is where CDN's are helpful. CDN's are edge servers which store lot of commonly accessed content and the CDN's are spread across the globe such that every end user has access to the nearest CDN.
For EX: Times of India may store the current day + one-week news, videos and published content in the CDN while the older archives can be stored at their centralized content hub's/data centers

**Access Logs:** Access logs on CDN's provide critical information on the type of request, device operating system which made the request, URL, IP address, time of the request, Brower type, amount of data downloaded etc. These logs can be mined for understanding the user persona, the device and operating systems these requests are made from and use them to contextualize the content to the end user

**S3 Bucket:** S3 buckets in AWS can be used to store object type of data. The advantages with using S3 buckets are

1. They are highly secure and access to it can be regulated using bucket policies
2. Scalable and Resilient with multiple redundant copies in different Availability zones
3. Provide data consistency – Consistent read and write
4. API Access – Supports REST API's

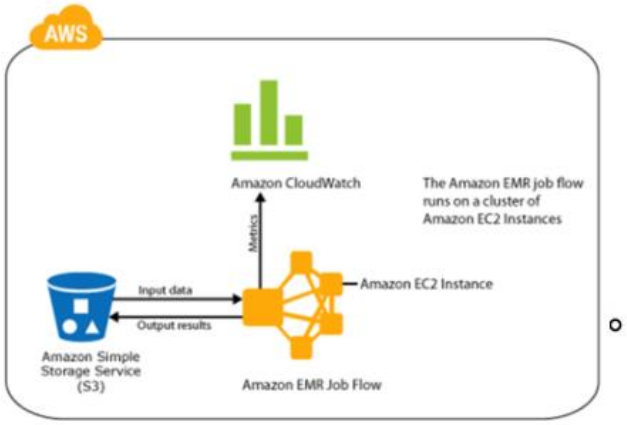
All the CDN logs and logs from any other access servers are pushed to a S3 bucket.
This can be done by creating scripts on the CDN Servers to push the logs to AWS Buckets. AWS credentials will need to be provided for performing this activity and adequate bucket policies will need to be set

**Elastic Map Reduce (EMR):** The core of this Big Data setup is the Elastic Map Reduce function from AWS
Amazon EMR provides a managed Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data. Key benefits of using an EMR are

- Easy to Use – Hadoop configuration is relatively easier than setting it up On-Premise
- Low cost – Pay per use by minutes of usage
- Elastic – EMR can scale to as many nodes as needed using AWS EC2
- Reliable – Hadoop cluster tuning is more efficient. Very little bug fixes
- Secure – EC2 firewall automatically setup. Security can be enforced on S3 buckets and EMR can also be setup in a VPC
- Flexible – Root access to every cluster server, bootstrap functions can easily be defined

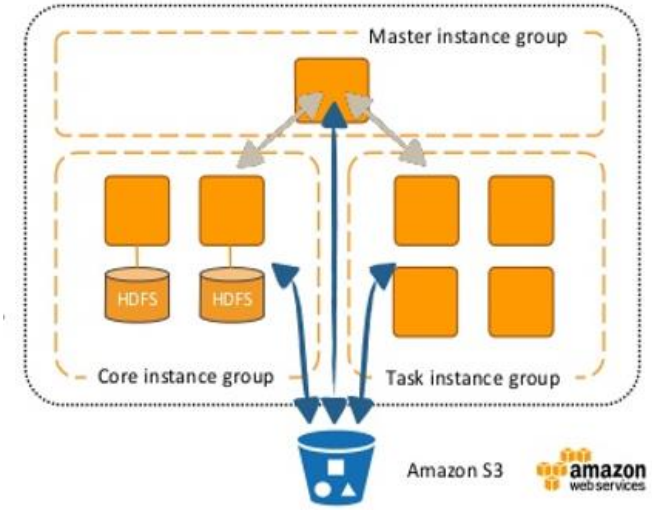
Below is the representation of the EMR architecture for a Big Data analytics for small sized (<2 PB data) data dump

*Figure 3.2 AWS EMR interaction with S3*




*Source: Amazon Web Services EMR Documentation*

*Figure 3.3 AWS EMR Master-Core Setup*




*Source: Amazon Web Services EMR Documentation*

The Amazon EMR The AWS EMR consists of the following layers each of which provides certain capabilities and functionality to the cluster

- Storage – HDFS and EMRFS, Local File Systems
- Cluster resource management: YARN
- Data processing Frameworks: Hadoop Mapreduce and Apache Spark
- Applications and Programs: JAVA, HIVE, PIG, SPARK SQL, HIVE SQL, Mlib, GraphX

The recommended solution for a small setup (<2 PB) data dump would consist of

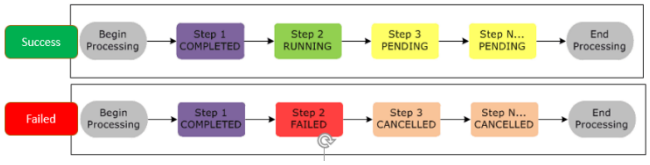
1. Master Node – m5. xlarge with 4 vCPU and 16 Gb Memory running Red hat Linux – 1 no
2. Core Node – c5. large with 2 vCPU and 4 Gb Memory running Red hat Linux – 3 no's
3. Task node - c5. large with 2 vCPU and 4 Gb Memory running Red hat Linux – 2 no's running in HA mode

Executing on EMR Cluster:

Following is the sequence of steps that will be executed on the EMR cluster when the map reduce job is fed to the cluster

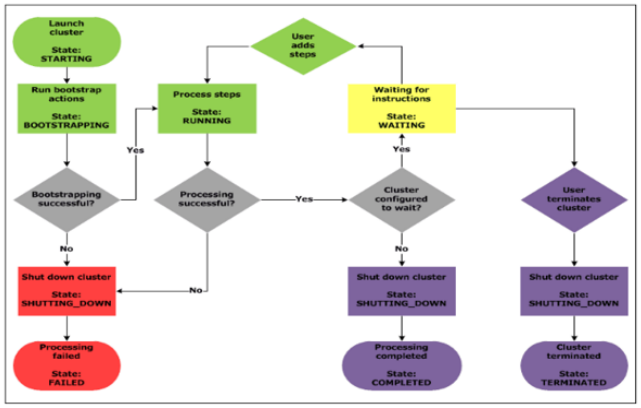
1. A request is submitted to begin processing steps.
2. The state of all steps is set to PENDING.
3. When the first step in the sequence starts, its state changes to RUNNING. The other steps remain in the PENDING state.
4. After the first step completes, its state changes to COMPLETED.
5. The next step in the sequence starts, and its state changes to RUNNING. When it completes, its state changes to COMPLETED.
6. This pattern repeats for each step until they all complete and processing ends.

*Figure 3.4 AWS EMR cluster life cycle*




The EMR cluster life cycle will be as shown below

*Figure 3.5 AWS EMR execution workflow*




In this setup we will provision the EMR Cluster using Hadoop or SPARK

Using Bootstrap actions we will install applications that will run on this cluster, in this case it will be a hive sql serialiser deserialiser function

The EMR cluster will need to be terminated Automatically once the entire job is completed. This will prevent unnecessary costs incurred on AWS

We will also need to enable DATA protection on abnormal cluster temination

EMR Scalability

Amazon EMR supports auto scalability for both task and core nodes. The Scaling policy for these nodes can be defined when configuring the instances at the launch of the instance

We can configure the auto scaling option using the AWS CLI function or the console It is important to configure both Scale-In and Scale-Out to esnsure that when the amount of data processing is large more task nodes are spun off and once the processing job is completed the nodes are scaled back in Identity and access managemnt (IAM) role for Autoscaling As autoscaling is an automatic function we need to setup a IAM role with the necessary permissions to add and terminate instances during auto scaling

We will need to run the following commands from the AWS Console for setting up the IAM role

1. *Create-default-roles*
2. *–auto-scaling-role EMR_AutoScaling_DefaultRole*

We will also need to define the Maximum and Minimum instances to limit the scale-out and scale-in nodes
Below is the contents of the AWS autoscale.json file where scale out of 10 nodes maximum and 2 nodes minimum is defined

```
"AutoScalingPolicy":
 {
  "Constraints":
  {
   "MinCapacity": 2,
   "MaxCapacity": 10
  },
  "Rules":
  [
  {
   "Name": "Default-scale-out",
   "Description": "Replicates the default scale-out rule in the console for YARN memory.",
   "Action":{
    "SimpleScalingPolicyConfiguration":{
     "AdjustmentType": "CHANGE_IN_CAPACITY",
     "ScalingAdjustment": 1,
     "CoolDown": 300
    }
   },
   "Trigger":{
    "CloudWatchAlarmDefinition":{
     "ComparisonOperator": "LESS_THAN",
     "EvaluationPeriods": 1,
     "MetricName": "YARNMemoryAvailablePercentage",
     "Namespace": "AWS/ElasticMapReduce",
     "Period": 300,
     "Threshold": 15,
     "Statistic": "AVERAGE",
     "Unit": "PERCENT",
     "Dimensions":[
      {
       "Key" : "JobFlowId",
       "Value" : "${emr.clusterId}"
      }
     ]
    }
   }
  }
 ]
}
```

**HIVE tables:** The data which is stored in the AWS S3 bucket – Log files are in text format and cannot be mined unless it is put in a proper format
HIVE SQL will be used for performing the data transformaiton.
As a part of this transformation following activities will need to be done

1. Use HIVE SCRIPT to create a HIVE table with appropriate Schema
2. Use the built-in regular expression serializer/deserializer (RegEx SerDe) to parse the input data and apply to the table schema
3. Run a HiveSQL query against the table and write the query results to the Amazon S3 output location specified
4. Various HIVESQL queries can be run to extract relevant data points needed to derive intelligence on end user usage
5. The HIVESQL queries can be enabled as part of bootstrap function and can be programmed to execute once the EMR cluster is in READY TO EXECUTE Mode

**Output S3 bucket**

The output S3 bucket is used to store the results generated by the HIVE SQL Query, this bucket will be secured through bucket policies and access is restricted
The output file stored in this bucket will depicit the different operating system, devices which have accessed the CDN. Based on the number of hits, the Media Publising company can determine the trends on the OS access and can create a plan to customize the content to match the device OS capabilities and form factors

## IV. RESULTS

Below is the total cost for running a 3 node cluster on AWS EMR
The total cost for 10 hours' usage is $4.74. This will be the cost to do analysis on a 2 PB data once a month

*Figure 4.1 AWS Usage and Costs view*

| | | |
|---|---|---|
| **Amazon Internet Services Private Ltd.** | | **$4.74** |
| CloudWatch | | $0.00 |
| Data Transfer | | $0.00 |
| Elastic Compute Cloud | | $3.30 |
| **US West (N. California)** | | **$3.30** |
| Amazon Elastic Compute Cloud running Linux/UNIX | | $3.30 |
| $0.308 per On Demand Linux m3.xlarge Instance Hour | 10.708 Hrs | $3.30 |
| EBS | | $0.00 |
| $0.00 per GB-month of General Purpose (SSD) provisioned storage under monthly free tier | 0.144 GB-Mo | $0.00 |
| Elastic MapReduce | | $0.72 |
| **US West (N. California)** | | **$0.72** |
| Amazon Elastic MapReduce USW1-BoxUsage:m3.xlarge | | $0.72 |
| $0.07 per hour for EMR m3.xlarge | 10.322 Hrs | $0.72 |

Below is the cost of setting up a Hadoop Big Data Cluster On-Premise with the assumption that all the software's are open source and don't require any licensing costs
As you can see the cost of running a Hadoop Cluster on EMR is a very small fraction vs the cost what would be needed to do a dedicated setup

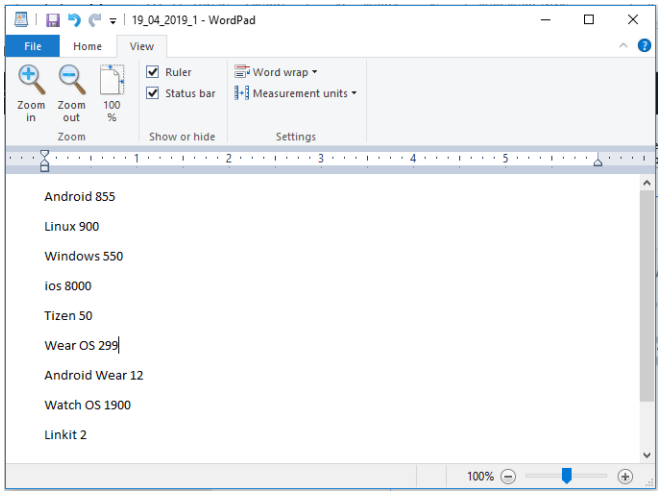
*Figure 4.2 Costs for Setting up On-Premise*

| Category | One Time | Y-1 | Y-2 | Y-3 | Y-4 | Y-5 |
|---|---|---|---|---|---|---|
| Compute | $ 15,000.00 | | | | | |
| Software | | $ 8,601.88 | $ 4,285.98 | $ 4,320.33 | $ 4,356.33 | $ 4,394.33 |
| Storage | $ 880,803.84 | | | | | |
| Hosting | $ 46,278.00 | $ 44,265.60 | $ 46,118.88 | $ 48,064.82 | $ 50,108.07 | $ 52,253.47 |
| Support costs | | $ 395,200.00 | $395,200.00 | $395,200.00 | $395,200.00 | $395,200.00 |
| **Total** | $ 942,081.84 | $ 448,067.48 | $445,604.86 | $447,585.15 | $449,664.39 | $451,847.79 |

| | | |
|---|---|---|
| Total Assets | $ | 1,208,851.51 |
| Total Support | $ | 1,976,000.00 |
| | $ | 3,184,851.51 |
| Per month cost | $ | 53,081 |

Below is the output file present in the Output S3 bucket post running an analytics exercise for a sample data
As you can see it clearly shows all the Operating Systems which have accessed the published content.
We can also note that there are references to multiple wearable Operating system which are relatively less but touted to grow exponentially

*Figure 4.3 Output of big data Analytics*




## V. CONCLUSION

Based on the findings of the commercial model it is very evident that running Big Data analytics on relatively small size of data on an in-frequent basis (Once a month) is very economically on AWS EMR vs running it on a dedicated setup in a data center
It can also be derived that the output from these analytics is very important for publishing companies to see the trend of devices accessing the content so that the Media and Publishing companies can create content to cater to this new emerging section of users who are accessing the Rich media content via Smart Phones, Wearables, Tablets and Phablets
We can also see that there are very few management overheads and better security options with the AWS cloud solution vs the On-Premise solution
Lastly, when it comes to scalability and elasticity, AWS cloud is far superior as it can scale up from a small sized data chunk to large data S3 buckets seamlessly which is not very easy On-Premise
This solution can be further implemented on larger data chunks for analytics and the economies of scale realized from AWS EMR Implementation can be determined and ascertained

## REFERENCES

1. Amazon EMR documentation - https://docs.aws.amazon.com/emr/index.html
2. HiveSQL - https://hive.apache.org
3. Google Consumer Barometer
4. Reuters Institute Digital News Report

## Authors Profile

**Girish G** holds a B.E degree in Informaiton Science and Engineering from Bangalore University. He has 20 years of Industry experience designing complex Infrastructure solutions for customers worldwide. His area of specialization is Data Center design, hybrid cloud solutions and remote infrastructure management. His areas of interest include PaaS and Public cloud solutions, Server less computing and Composable infrastructure

**Dr. M. Prabhakar** holds a Master's Degree & Ph. D. in Computer Engineering from Anna University, Chennai. He has 21 years of teaching experience, teaching various subjects for bother UG and PG Programs under various Universities. He is involved in research in the areas of Adhoc Networks, Wireless Sensor Networks, Cluster Computing, & Image Processing. He has published over 25 papers in National and International Journals and 8 papers in National and International journals.