CLOUD COMPUTING

Amazon Elastic Compute Cloud – EC2 Amazon Elastic MapReduce – EMR

Leonidas Akritidis – lakritidis@ihu.gr

Introduction to Amazon EC2

EC2 stands for Elastic Compute Cloud.

- A cloud service that provides secure and resizable compute capacity in the cloud.
 - In simple words: flexible processing power on demand.
- Designed to make web-scale cloud computing easier for developers.
- SLA commitment of 99.99% availability for each Amazon EC2 region. Each region consists of at least 3 availability zones.

EC2 Instances

- The services of EC2 are provided through the creation, deployment, and usage of EC2 instances.
- An instance is simply a virtual computing environment created with the aim of executing of a particular job.
- An instance type is a configuration of CPU, memory, storage, and networking capacity of an instance.
- There are 275 such instance types to help optimize the cost and performance of workloads.
- Available with choice of CPU, storage and networking options, operating system, and purchase model.

EC2 Instance Categories

- General Purpose: Ideal for business critical applications, small and mid-sized databases, web tier apps, etc.
- Compute Optimized: Ideal for high performance computing, batch processing, video encoding, and more.
- Memory Optimized: Ideal for high performance databases, distributed web scale in-memory caches, real time big data analytics, and more.
- Storage Optimized: Ideal for NoSQL databases, data warehousing, distributed file systems, and more.
- Accelerated Computing: Ideal for machine learning, graphic intensive applications, gaming, and more.

EC2 Instance Types and support

- More information and full details on the provided hardware per instance type <u>https://aws.amazon.com/ec2/instance-types/</u>
- Note: Some instance types are not supported by some availability zones (e.g. <u>slide 14</u>).
- If an instance type is not supported by the current availability zone of the user, then we should either:
 - Change the availability zone/region, or
 - Select another instance type.

EMR: Amazon Elastic MapReduce

- Amazon offers the ability to create MapReduce clusters and deploy standard MapReduce jobs on these clusters, through its EC2 infrastructure.
- □ This service is called **Elastic MapReduce (EMR)**.
- □ It resides inside the "Analytics" group of services, within the main menu of AWS management console.
- EMR launches EC2 instances that serve as MapReduce processing nodes and HDFS data nodes.
- □ <u>Supported instance types for EMR</u>.

EMR Work flow (Step 2)

1. Create a bucket in Amazon S3. 🥝

- 2. Create an Amazon EC2 key pair for securely accessing the master node of the EMR cluster.
- 3. Create and configure an EMR cluster in EC2.
- 4. Study the problem and design a MapReduce algorithm.
- 5. Upload the input data to be processed and the executable code in a S3 bucket.
- 6. Deploy the job across the cluster.
- 7. Retrieve the output of the application in a S3 bucket.

EC2 Key Pairs (1)

- Key pairs constitute a strategy for securely accessing and managing EC2 resources (in our case, the EMR cluster).
- The are used in replacement of the conventional "log-in" procedure through usernames and passwords.
- Key pair = Two keys: one held by Amazon (public key) and one held by the user (private key).
- The communication between the two parts (Amazon and the user) is performed by exchanging encrypted messages.

Public/Private keys – PEM/PPK files

- The public and private keys are used to encrypt a message upon transmission, and decrypt the cipher upon receipt.
 - A sniffer who steals an encrypted message cannot decrypt it. Its contents are inaccessible without the private key.
 - A sniffer cannot pretend that he/she is any of the two communicating parts.
- A PEM or a PPK file will be created and associated with every key pair that is created by the user.
- One of) These files are required to contact directly the master node of an EMR cluster via SSH.

Creating an EC2 Key Pair

- Key pairs are created from AWS management console.
- □ From the left-handed pane, select "NETWORK & SECURITY → Key Pairs".
- □ Then, press the "Create Key Pair" button.
- Set a name for the key-pair.
- Select ppk File format.
- Download and store the generated ppk file.
- This file will later grant SSH access to the EMR cluster via PuTTY.



EMR Work flow (Step 3)

- 1. Create a bucket in Amazon S3. 🥝
- 2. Create an Amazon EC2 key pair for securely accessing the master node of the EMR cluster.
- 3. Create and configure an EMR cluster in EC2.
- 4. Study the problem and design a MapReduce algorithm.
- 5. Upload the input data to be processed and the executable code in a S3 bucket.
- 6. Deploy the job across the cluster.
- 7. Retrieve the output of the application in a S3 bucket.

EMR & AWS Management Console

aws Servic	es 👻 Resource Groups 👻	۶	
Amazon EMR Clusters Security configurations	Welcome to Ama Amazon Elastic MapReduce (Am analysts and developers to easil	azon Elastic MapR	educe
Block public access VPC subnets Events Notebooks Git repositories	You do not appear to have any cl Create cluster How Elastic MapRedu	lusters. Create one now:	nounts of data.
Help What's new	Upload	Create	Monitor
	Upload your data and processing application to S3.	Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.	Monitor the health and progress of your cluster. Retrieve the output in S3.

EMR: Creating a MapReduce Cluster

- 1. S3 bucket for MapReduce I/O.
- Hadoop MapReduce and accompanying applications (<u>Apache Hive data warehouse</u> <u>storage system</u>, <u>Mahout distributec</u> <u>linear algebra framework</u>, etc).
- 3. The instance type determines the hardware specifications of the machines of the cluster.
- Number of instances represent the number of machines in EMR Cluster (must be > 2, since 1 is reserved for the Master).
- 5. EC2 key pair for accessing the EMR cluster via SSH.

	General Configuration		
	Cluster name	ihu-cluster Logging S3 folder s3://ihu-bucket/	
	Launch mode	Cluster Step execution	
	Software configuration		
	Release	emr-5.29.0	0
-	Applications	Core Hadoop: Hadoop 2.8.5 with Ganglia 3.7.2, Hive 2.3.6, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2	
		HBase: HBase 1.4.10 with Ganglia 3.7.2, Hadoop 2.8.5, Hive 2.3.6, Hue 4.4.0, Phoenix 4.14.3, and ZooKeeper 3.4.14	
		Presto: Presto 0.227 with Hadoop 2.8.5 HDFS and Hive 2.3.6 Metastore	
		Spark: Spark 2.4.4 on Hadoop 2.8.5 YARN with Ganglia 3.7.2 and Zeppelin 0.8.2	
		Use AWS Glue Data Catalog for table metadata	0
	Hardware configuration		
	B Instance type	m5.xlarge	The selected instance type adds 64 GiB of GP2 EBS
	4 Number of instances	3 (1 master and 2 core nodes)	Storage per instance by default. Learn nore 🗗
	Security and access		
	5 EC2 key pair	ihu-keypair	Learn how to create an EC2 key pair.
Ç	Permissions	Default Custom	
		Use default IAM roles. If roles are not present, they will be created for you with managed policies for automatic policies	be automatically icy updates.
	EMR role	EMR_DefaultRole 🖸 🚯	

EC2 instance profile EMR_EC2_DefaultRole [2]

EMR Cluster: Instance types

Default instance type: m5.xlarge.

A general purpose instance "provides a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads."

General Purpose	General Purpose											
Compute Optimized	seneral purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse vorkloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code											
Memory Optimized	repositories.											
Accelerated Computing	A1 T3 T3a T2 M6g M5 M5a M5n M4											
storage Optimized	M5 instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175M processors. This											
nstance Features	family provides a balance of compute, memory, and network resources, and is a good choice for many applications.											
Measuring Instance	Features:											
Performance	• Up to 3.1 GHz Intel Xeon® Platinum 8175M processors with new Intel Advanced Vector Extension (AVX-512) instruction set											
	 New larger instance size, m5.24xlarge, offering 96 vCPUs and 384 GiB of memory 											
	Up to 25 Gbps network bandwidth using Enhanced Networking											
	Requires HVM AMIs that include drivers for ENA and NVMe											
	Powered by the AWS Nitro System, a combination of dedicated hardware and lightweight hypervisor											
	 Instance storage offered via EBS or NVMe SSDs that are physically attached to the host server 											
	• With M5d instances, local NVMe-based SSDs are physically connected to the host server and provide block-level storage that is coupled to the lifetime of the M5 instance											
	New 8xlarge and 16xlarge sizes now available.											

EMR Cluster: Instance type hardware

Details for the hardware specifications of the m5.xlarge instances can be found under the M5 tab.

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
m5.large	2	8	EBS-Only	Up to 10	Up to 4,750
m5.xlarge	4	16	EBS-Only	Up to 10	Up to 4,750
m5.2xlarge	8	32	EBS-Only	Up to 10	Up to 4,750
m5.4xlarge	16	64	EBS-Only	Up to 10	4,750
m5.8xlarge	32	128	EBS Only	10	6,800
m5.12xlarge	48	192	EBS-Only	10	9,500
m5.16xlarge	64	256	EBS Only	20	13,600
m5.24xlarge	96	384	EBS-Only	25	19,000
m5.metal	96*	384	EBS-Only	25	19,000
m5d.large	2	8	1 x 75 NVMe SSD	Up to 10	Up to 4,750
m5d.xlarge	4	16	1 x 150 NVMe SSD	Up to 10	Up to 4,750
m5d.2xlarge	8	32	1 x 300 NVMe SSD	Up to 10	Up to 4,750

Note: m5.xlarge may not be supported by the current availability zone (in this case, see again <u>slide 5</u>).

Managing the EMR Cluster

- With these simple actions, AWS will create and start an EMR cluster with 3 nodes (that is, 3 EC2 instances).
- The cluster can be cloned or terminated at any time from the buttons at the top.
- The management and monitoring of the cluster are performed from the tab headers at the top of the screen.

Clone Terminate	AWS CLI export						
Cluster: ihu-cluste	r Running Running step	D					
Summary Application	on history Monitoring	Hardware	Configurations	Events	Steps		
Connections:	Enable Web Connection -	- Hue, Ganglia,	Resource Manager .	(View All)			
Master public DNS:	100.27.4.65 SSH						
History service:							
Tags:	View All / Edit						
Summary		Configura	tion details				
ID: j-O	TFSPEMIG970	Rel	ease label: emr-5.29	0.0			
Creation date: 202 Elapsed time: 8 m	20-03-16 00:16 (UTC+2) inutes	di	Hadoop Amazon : stribution:	2.8.5			
After last step Clu completes:	ster waits	Applications: Ganglia 3.7.2, Hive 2.3.6, Hue 4.4.0, Mahout 0.13.0, Pig 0.17.0, Tez 0.9.2					
Termination Off	Change		Log URI: s3://aws-logs-407705302836-us- east-1/elasticmapreduce/				
protection		EMRFS	consistent Disabled view:				
		Cust	om AMI ID:				
Network and hardware		Security a	and access				
Availability zone: us-	east-1e		Key name: ihu-keyp	air			
Subnet ID: sub Master: Boo	net-0de8befa74baab235 🔀	EC2 instance EMR_EC2_DefaultRole profile:					
Core: Provisioning 2 m3.xlarge			EMR role: EMR_DefaultRole				
Task:	• •	Visible to	o all users: All Char	<u>ige</u>			
		Security	groups for sg-0e974 Master: (ElasticM	47866507580 lapReduce-m	de2 🛂 laster)		
		Security	groups for sg-09c9a	a74d5e23c16	674 🔼		

Managing the EMR Cluster (2)

- IP address to connect with SSH.
 Note: SSH Connection to the Master node of the cluster is not allowed by default. Several configuration options must be set before SSH connection is possible. See next.
- 2. EMR Cluster ID, creation date and uptime.
- 3. Master and Worker nodes general status. Subnet ID is crucial to access the VPC which hosts the cluster.
- 4. Security settings: active key pair, security groups and permissions for the Master and Worker nodes of the cluster.



Connecting to EMR Cluster via SSH (1)

- Slides 18 25 present the necessary configuration settings for accessing our EMR cluster via SSH.
- SSH (Secure Shell) is a protocol that establishes secure network connections over an encrypted channel (in contrast to Telnet where the connections are unencrypted).
- It is a command line tool that allows secure remote access to a network resource, by adopting the well-known clientserver architecture.
- We shall utilize PuTTY, a popular, open-source, SSH/Telnet client.

Connecting to EMR Cluster via SSH (2)

- For each one of the nodes of the EMR cluster, AWS automatically creates and launches an EC2 instance.
- These instances in turn, reside into Virtual Private Clouds (VPCs).
- To allow access to the Master of the EMR cluster, both the VPC and the security group of the respective EC2 instance, must be granted the appropriate permissions.
- The checklist in the following slide contains the necessary steps to achieve this goal.

Configurations for SSH Connection

To establish a SSH connection with the EMR cluster, the steps of the following checklist must be completed.

- 1. Configure the VPC (Virtual Private Cloud) which hosts the EMR cluster.
- 2. Locate the EC2 instance which hosts EMR Master.
- 3. Grant the appropriate permissions for SSH access to the "ElasticMapReduce-master" security group.
- 4. Configuring PuTTY for establishing a SSH connection with the EMR Cluster.

Configuring the VPC

Step 1: configure the VPC (Virtual Private Cloud) which hosts the EMR cluster.

- 1. From the previous slide, on ⁶, click on the Subnet ID.
- 2. Click on the VPC of the Subnet.
- 3. From the Actions button, make sure that "DNS Resolution", "DNS Hostnames", and "ClassicLink DNS Support" are all enabled.



Configuring the EC2 instance

Step 2: locate the EC2 instance which hosts EMR Master.

- 1. From AWS Management Console, select "EC2".
- 2. In the EC2 Dashboard, click on the "Running Instances: 3" link.
- 3. On the list with the three EC2, instances, locate the one which has the security group "*ElasticMapReduce-master*". Click on the respective link.

aws Services ^	Resource Groups													
History	Find a service by r	Launch Instanc	e 🔻 Conne	Actions V	/							₫	⊕ ♦	0
Console Home	<u> </u>	Q Filter by tags	and attributes or se	arch by keyword							⊘ _ K <	1 to 3	of 3 >	>
EC2	Compute	Name	✓ Inst	ance ID 🖌	Instance Type 👻	Availability Zone -	Instance State	- Status Checks -	Alarm Statu	15	Public DNS (IPv4)	- IPv4	Public IP	~
VPC	EC2		i-000	ce99448d4d84fc5	m3.xlarge	us-east-1e	running	2/2 checks	None	6	ec2-100-27-4-65.co	100.2	27.4.65	
			i-019	9bdfe9a6949a42d	m3.xlarge	us-east-1e	running	2/2 checks	None	6	ec2-100-24-255-227	100.2	24.255.227	
EC2			i-090	d7f666a9217de59	m3.xlarge	us-east-1e	running	2/2 checks	None	\ @	ec2-52-73-253-86.c	52.73	3.253.86	
		•												Þ
Resources		Instance: i-000	e99448d4d84fc5	Public DNS	: ec2-100-27-4-65.c	ompute-1.amazonaw	vs.com						886	3 🔺
You are using the following Amazon I	EC2 resources in the US East (Description	Status Checks	Monitoring	Tags									
Running instances	3		Instance ID	i-00ce99448d4d8	4fc5			Public DNS (IPv4) e	2-100-27-4-65	.compu	ite-1.amazonaws.com			
Dedicated Hosts	0		Instance state	running				IPv4 Public IP 1	00.27.4.65					
			Instance type	m3.xlarge				IPv6 IPs -						
Volumes	3		Finding	Opt-in to AWS C recommendations	ompute Optimizer for s. Learn more			Elastic IPs						
Key pairs	1		Private DNS	ip-172-30-0-163.e	ec2.internal			Availability zon	s-east-1e					
Placement groups	0		Private IPs	172.30.0.163				Security groups	lasticMapRedu	ce-mas	ster. view inbound rules.	view		

Configuring the Security Group

Step 3: grant the permissions for SSH access to the "ElasticMapReduce-master" security group.

- 1. From the "Actions" button, select "Edit inbound rules".
- 2. In the list of rules that opens, click on the "Add rule" button at the bottom of the list.
- 3. Add a rule for SSH access to anybody, as shown on the figure.
- 4. Click on the "Save rules" button.

EC2 > Security Groups	
Security Groups (1/1) Info	C Actions A Create security group
Q Filter security groups	View details
Security group ID: ca.0e07/786650758de2	Edit inbound rules
clear mers	Edit outbound rules
Security group ID 🔺 Security group name \triangledown VPC ID \triangledown Description \heartsuit	Owner V Inbound rules of Manage tags
sg-0e974786650758de2 ElasticMapReduce-mas vpc-080596ce15d8914b8 🖸 Master group for Elasti	407705302836 20 Permission entries 1 Permission entry
3 SSH ▼ TCP 22	Custom 🔺 Q
	Custom
2 Add rule	Anywhere
	My IP Anywhere

Connecting to the EMR Cluster via SSH with PuTTY

Step 4: Now we are ready to use the popular, freeware, third party software, named PuTTY, to establish a SSH connection to the cluster.

1. Download PuTTY.exe to your computer from:

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

- 2. Start PuTTY.
- 3. In the Category list, click Session.
- 4. In the Host Name field, type the correct hostname (<u>slide 16</u>, field 1).
- 5. In the Category list, expand "Connection \rightarrow SSH", and then click "Auth".
- 6. For Private key file for authentication, click Browse and select the private key file (**ihu-keypair.ppk**) used to launch the cluster.
- 7. Click Open.
- 8. Click Yes to dismiss the security alert.

Connecting to the EMR Cluster via SSH with PuTTY

🔀 PuTTY Configuration	? ×
Putrty Configuration Category: Session Copy: Session Copy: Session Copy: Session Copy: Session Selection Selection Colours Selection Colours Colours Colours Selection Col	Pasic options for your PuTTY session Specify the destination you want to connect to Host Name (or IP address) Port hadoop@ec2-100-27-4-65.compute-1.arr 22 Connection type: Rag Cannection type: SSH Cannection type: SSH Cannection type: Rlogin Cannection type: SSH Cada, save or delete a stored session Saved Sessions Amazon IHU EMR Default Settings Amazon Problemia Old Amazon Problemia Delete Delete Close window on exit: C Always Never
About <u>H</u> elp	<u>Open</u> <u>Cancel</u>



🛃 hadoop@ip-172-30-0-163	3:~					<u> </u>
_ _ _) _ (/ \ _	Amazon Li	nux AMI				
https://aws.amazon.cc 20 package(s) needed Run "sudo yum update"	om/amazon- for secur 'to apply	linux-ami/2 sity, out of all update	2018.03-1 E 37 avai es.	release-not ilable	tes/	
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE	MARABANA M:	1 M: M M:: M M::: M M::: M M:::M M:::M M:::M M:::M M:::M M:::M	Imperimental Imperimental </td <td>RRRRRRRRR R:::::R R::::R R:::R R:::R R:::R R:::R R:::R R:::R R:::R R:::R R R:::R R R:::R R R R R R R R R R R R R R R R R R R R</td> <td>RRRRRR RR::::R R::::R R::::R R::::R RR:::R RR:::R RR:::R R::::R R::::R R::::R R::::R R::::R R::::R R::::R</td> <td></td>	RRRRRRRRR R:::::R R::::R R:::R R:::R R:::R R:::R R:::R R:::R R:::R R:::R R R:::R R R:::R R R R R R R R R R R R R R R R R R R R	RRRRRR RR::::R R::::R R::::R R::::R RR:::R RR:::R RR:::R R::::R R::::R R::::R R::::R R::::R R::::R R::::R	
[hadoop@ip-172-30-0-1	163 ~]\$					-

EMR Work flow (Step 4)

- 1. Create a bucket in Amazon S3. 🥝
- 2. Create an Amazon EC2 key pair for securely accessing the master node of the EMR cluster.
- 3. Create and configure an EMR cluster in EC2. 🥩
- 4. Study the problem and design a MapReduce algorithm.
- 5. Upload the input data to be processed and the executable code in a S3 bucket.
- 6. Deploy the job across the cluster.
- 7. Retrieve the output of the application in a S3 bucket.

Preparing the MapReduce job

- At this point the problem to be solved and the algorithm to be executed become the most important elements.
- We will compute Scientometrics in parallel by using MapReduce.
- Relevant article: L. Akritidis, P. Bozanis, "Computing Scientometrics in Large-Scale Academic Search Engines with MapReduce", In Proceedings of the 13th International Conference on Web Information System Engineering (WISE), Lecture Notes in Computer Science (LLNCS), vol. 7651, pp. 609-623, 2012.

Scientometrics

- Metrics evaluating the research work of a scientist by assigning impact scores to his/her articles.
- Usually expressed as definitions of the form:
 - A scientist a is of value V, if at least V of his articles have been assigned a score $S \ge V$.
- A researcher must author numerous qualitative and influential articles.
- □ Most popular metric: h-index, defined as,
 - □ A scientist *a* has h-index *h*, if at least *h* of his articles have received *h* citations (i.e., a score $S \ge h$).

h-index example

TITLE 🖪 : C	ITED BY	Article #	Citation Count	h-index
Identifying the productive and influential bloggers in a community L Akritidis, D Katsaros, P Bozanis IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and	61	1	6 1 ≥ 1	1
Identifying influential bloggers: Time does matter L Akritidis, D Katsaros, P Bozanis 2009 IFERMICIACM International Joint Conference on Web Intelligence and	59	2	$59 \ge 2$	2
The <i>f</i> index: Quantifying the impact of coterminal citations on scientists' ranking D Katsaros, L Akritidis, P Bozanis	40	3	$40 \ge 3$	3
Journal of the American Society for Information Science and Technology 60 (5 Effective rank aggregation for metasearching L Akritidis, D Katsaros, P Bozanis	36	4	36 ≥ 4	4
Journal of Systems and Software 84 (1), 130-143 Effective ranking fusion methods for personalized metasearch engines L Akritidis, D Katsaros, P Bozanis 2008 Panelolaris Conference on Information, 39.43	13	5	$13 \ge 5$	5
A supervised machine learning classification algorithm for research articles L Akritidis, P Bozanis Proceedings of the 28th Annual ACM Symposium on Applied Computing, 115-120	11	6	11 ≥ 6	6
Improved retrieval effectiveness by efficient combination of term proximity and zone scoring: A simulation-based evaluation L Akritidis, D Katsaros, P Bozanis Simulation Modelling Practice and Theory 22, 74-91	11	7	11 ≥ 7	7
Improving opinionated blog retrieval effectiveness with quality measures and tempora features L Akritidis, P Bozanis World Wide Web 17 (4), 777-798	il 10	8	$10 \ge 8$	8
Identifying attractive research fields for new scientists L Akritidis, D Katsaros, P Bozanis Scientometrics 91 (3), 869-894	10	9	$10 \ge 9$	9
Computing scientometrics in large-scale academic search engines with mapreduce L Akritidis, P Bozanis International Conference on Web Information Systems Engineering, 609-623	6	10	6 ≤ 10	9
Positional data organization and compression in web inverted indexes L Akritidis, P Bozanis International Conference on Database and Expert Systems Applications, 422-429	6			

QuadSearch: A novel metasearch engine

I Akritidie G Vouteakalie D Katearne P Rozanie

h-index in large-scale data

- To compute h-index, it is required that for each scientist we maintain a list of all of his/her articles sorted in decreasing order of citations.
- In large-scale data, there are numerous authors and numerous such lists.
 - CiteSeerX dataset: 10 million articles.
 - Microsoft Academic Graph: 209M articles, 253M authors.
- The required data does not fit into the main memory of a single workstation.
- □ h-index calculation must be performed in parallel.

Parallelizing the problem

- Goal: Compute Scientometrics in parallel
- □ Input: $(p_i, C_{pi}) \rightarrow (paperID, paperContent)$
- □ Output: $(a, h_a) \rightarrow (author, hindex)$
- To reach our goal, we have to construct for each author, a list of his/her articles sorted by decreasing number of citations:

$$\left(a, SortedList\left[\left(p_{1}, S_{x}^{p_{1}}\right), \left(p_{2}, S_{x}^{p_{2}}\right), ..., \left(p_{N}, S_{x}^{p_{N}}\right)\right]\right)$$

Then, we just iterate through the list and we compute the desired h-index value.



- Download a <u>draft preprint of the article</u>.
- □ A presentation of the article in WISE 2012.
- □ A <u>public GitHub repository</u> with:
 - The source code of the MapReduce algorithms in Java, and
 - a toy dataset with 100 articles from CiteSeerX.
- □ The <u>full CiteSeerX dataset</u>.

EMR Work flow (Step 5)

- 1. Create a bucket in Amazon S3. 🥝
- 2. Create an Amazon EC2 key pair for securely accessing the master node of the EMR cluster.
- 3. Create and configure an EMR cluster in EC2. 🥩
- 4. Study the problem and design a MapReduce algorithm.
- 5. Upload the input data to be processed and the executable code in a S3 bucket.
- 6. Deploy the job across the cluster.
- 7. Retrieve the output of the application in a S3 bucket.

Code & dataset

Download the code from the aforementioned <u>GitHub</u> repository.

- Build, compile and generate an executable (binary) JAR file from the downloaded code.
 - Some MapReduce, HDFS, YARN and other JARS maybe required on the build path of JAR.
 - Alternatively, advanced users can <u>compile the code and</u> <u>generate an executable binary JAR</u> through SSH connection with the Master machine of their EMR cluster.
- Download the toy dataset again from the aforementioned GitHub repository.

S3 ihu-bucket snapshots

ihu-buo	cket											
Over	view Proper	rties Pe	rmissions	Management	Access points							
]							ihu-bucket					
Q Type	e a prefix and press E	nter to search. P	ress ESC to clear.				Overview		dataset fol	der		
]					
🛓 Uploa	d + Create folde	er Download	Actions ~		US Ea	st (N. Virgi	r Q Type a pref	fix and press Enter	r to search. Press ESC to clear.			
						Viewing	1		Download Actions			
Nai	me 🔻			Last modified -	Size 🗸	Storage	C Upload		Download Actions ~		US Eas	st (N. Virginia)
	dataset											Viewing 1 to 1
	jars						Name 🗸			Last modified 👻	Size 🔻	Storage class -
				Mar 5, 2020 6:00:17 PM			🖹 csx_10	00.txt		Mar 16, 2020 7:07:04 PM GMT+0200	1.9 MB	Standard
	aire.pdf			GMT+0200	3.7 MB	Standard	1					
						< Viewing f	1 to 3 🔿					
ihu-bu	cket	iar	e foldo	r								
Over	view	jar										
								_				
Q Тур	e a prefix and press E	Enter to search. F	Press ESC to clear.									
🔔 Uploa	ad 🕂 Create fold	ler Downloa	Actions ~		US E	East (N. Vir	rginia) 🟾 😂					
						Viewin	a 1 to 2					
Na	me 🗸			Last modified -	Size 🗸	Storag	je class 🗸					
				Mar 18, 2020 3:31:51	00.01/5	01						
L á	wise2012.jar			AM GMT+0200	23.3 KB	Standa	ard					

EMR Work flow (Step 6)

- 1. Create a bucket in Amazon S3. 🥝
- 2. Create an Amazon EC2 key pair for securely accessing the master node of the EMR cluster.
- 3. Create and configure an EMR cluster in EC2. 🥩
- 4. Study the problem and design a MapReduce algorithm.
- 5. Upload the input data to be processed and the executable code in a S3 bucket.
- 6. Deploy the job across the cluster.
- 7. Retrieve the output of the application in a S3 bucket.

EMR Steps

- Open the management console and navigate to the created EMR cluster.
- The "Steps" tab accommodates the interface for creating, cancelling and monitoring the MapReduce jobs that have been scheduled on this cluster.
- AWS includes several types of such steps.
 - Running a standard & simplistic Word Count example job.
 - Running a custom JAR file,

🗖 etc.

EMR Steps management (1)

Cluster: ih		unning step										
Summary	Application history Monito	oring Hardware Configura	tions	ts Steps Bootstrap acti	ons							
Concurrency: 1 Change												
After last step	completes: Cluster waits											
2 Add step	Clone step Cancel step	0										
Brilter: All	steps	7 steps (all lo	aded) C									
	ID	Name	4 Status	Start time (UTC+2) 🚽	Elapsed time	5Log files 🖸						
	s-2TXDYR2SULALO	Job_WISE_2012	Running	2020-03-18 02:38 (UTC+2)	45 seconds	View logs						
● ▶ ▲	s-1DLK9YHS0FYGV	Job_WISE_2012	Failed	2020-03-18 02:34 (UTC+2)	30 seconds	No logs created yet C						
● ▶ ▲	s-1EIS400XS1V8Q	Custom JAR	Failed	2020-03-18 02:18 (UTC+2)	20 seconds	controller syslog stderr stdout* $f C$						
● ►	s-Z8PP9A8U0C9S	Custom JAR	Cancelled			View logs						
● ►	s-1MAXPAIW2QP2B	Custom JAR	Cancelled			View logs						
● ►	s-0QI9AXL131I0	Custom JAR	Cancelled			View logs						
● ►	s-1SNDQHTE24Z00	Setup hadoop debugging	Completed	2020-03-18 02:11 (UTC+2)	2 seconds	View logs						

EMR Steps management (2)

- Cluster Status: "Waiting" (cluster is ready to accept new jobs), "Running" (cluster is running the scheduled step/s), "Terminated".
- 2. Add a new step button.
- 3. List of scheduled steps. The pending steps will be run sequentially unless the user cancels some of them.
- 4. Step status (pending, running, failed, completed, success).
- 5. Log files. In most cases *stderr* is the log file of interest. It enlists Java & MapReduce runtime errors.
- □ Click on "Add Step".

Creating a Step in EMR cluster (1)

- □ A dialog box for the creation of a new step appears.
- □ Step type: Custom JAR.
- □ Name: A custom name.
- JAR location: the location of the executable JAR file within a valid S3 bucket.
- The arguments are application-specific.
- In our case, they reflect the input & folders in S3.

dd step			×
Step type	Custom JAR	~	
Name*	Job_WISE_2012		
JAR location*	s3://ihu-bucket/jars/wise2012.jar		JAR location maybe a path into S3 or a fully qualified java class in the classpath.
Arguments	s3n://ihu-bucket/dataset/csx_100.txt s3n://ihu-bucket/output 1		These are passed to the main function in the JAR. If the JAR does not specify a main class in its manifest file you can specify another class name as the first argument.
Action on failure	Continue	~	What happens if the step fails
			Cancel Add

EMR Steps management (2)

- Click on the "Add" button.
- The new job is appended in the "Steps" list with status "Pending".
- The step will be executed automatically and its status will change according to its success/failure.
- On the latter case, the log files provide valuable information (especially the stderr one).

EMR Work flow (Step 6)

- 1. Create a bucket in Amazon S3. 🥝
- 2. Create an Amazon EC2 key pair for securely accessing the master node of the EMR cluster.
- 3. Create and configure an EMR cluster in EC2. 🥩
- 4. Study the problem and design a MapReduce algorithm.
- 5. Upload the input data to be processed and the executable code in a S3 bucket.
- 6. Deploy the job across the cluster. 🥩
- 7. Retrieve the output of the application in a S3 bucket.

Collecting the job's output

- In the case of successful completion, EMR will automatically create a folder named "output" inside our ihu-bucket.
- \square This folder contains the files with the output of the job.
- □ In our example, only one such file exists, part-r-00000.
- For larger jobs with multiple Reducers, the output may span multiple files.

Collecting the job's output



Thank you! Any Questions?