



AWS Summit

AWS技术峰会 2015 · 上海





Amazon EMR(Elastic MapReduce)技术分享

韩小勇



什么是EMR(Elastic MapReduce)

- AWS上的Hadoop生态系统



为什么使用EMR



易于使用

在几分钟内创建一个集群



成本低廉

按小时付费/预留实例/竞价实例



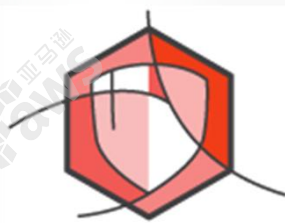
弹性

轻松增加或减少容量



可靠

监控集群，重试失败任务，自动替换性能不佳实例



安全

防火墙/隔离的网络/文件加密

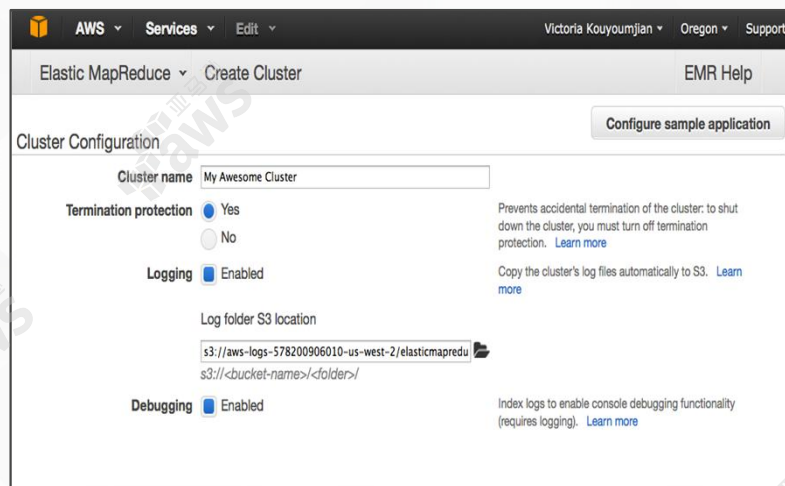


灵活

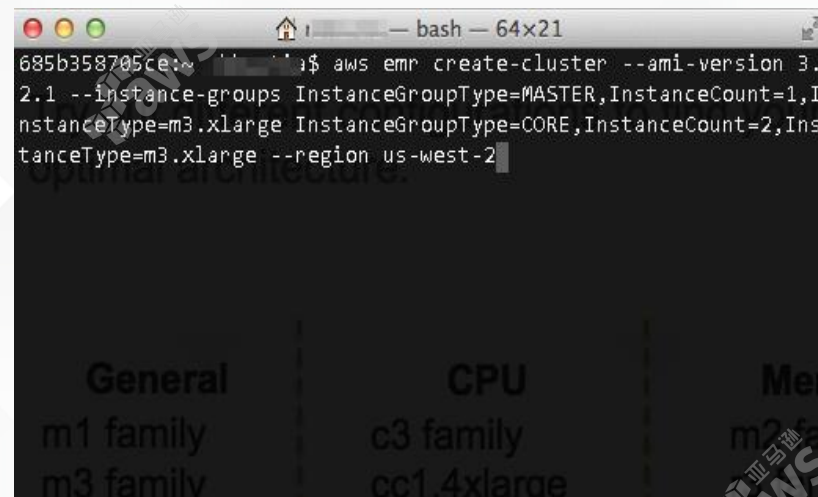
轻松安装额外应用程序和定制每个集群

部署简单

AWS控制台



AWS命令行



或者使用您习惯的SDK：

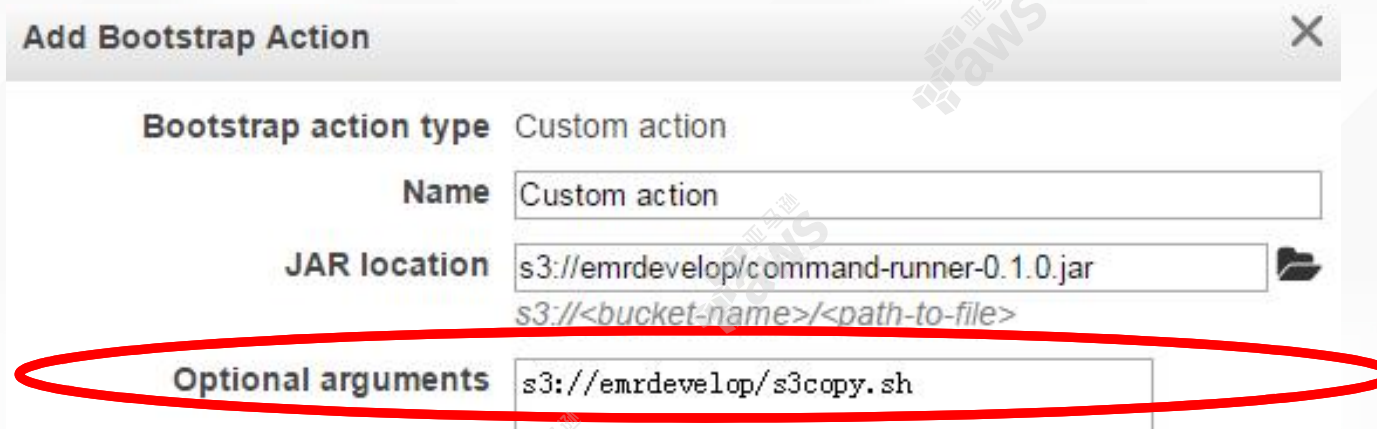


.....



安装灵活

使用Bootstrap安装更多的应用



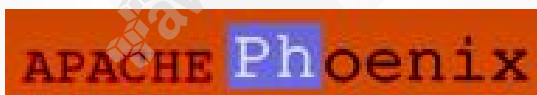
Add Bootstrap Action

Bootstrap action type: Custom action

Name: Custom action

JAR location: s3://emrdevelop/command-runner-0.1.0.jar
s3://<bucket-name>/<path-to-file>

Optional arguments: s3://emrdevelop/s3copy.sh



<https://github.com/aws-labs/emr-bootstrap-actions>

实例多样

使用不同的实例构造最优的架构

通用型

m1系列
m3系列

批处理

CPU密集

c3系列
cc1.4xlarge
cc2.8xlarge

机器学习

内存密集

m2系列
r3系列

Spark以及
交互式应用

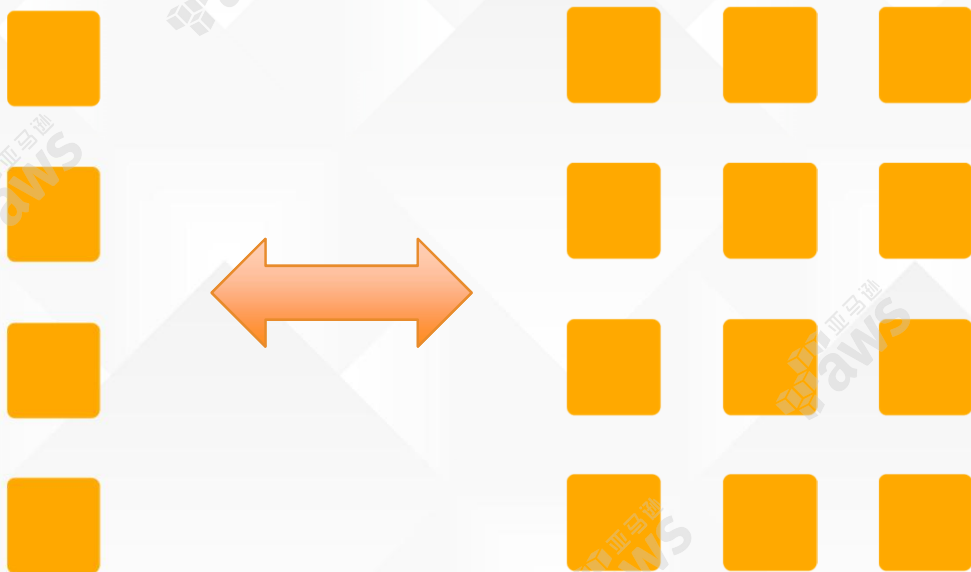
磁盘及IO

d2系列
i2系列

大容量HDFS

容量可变

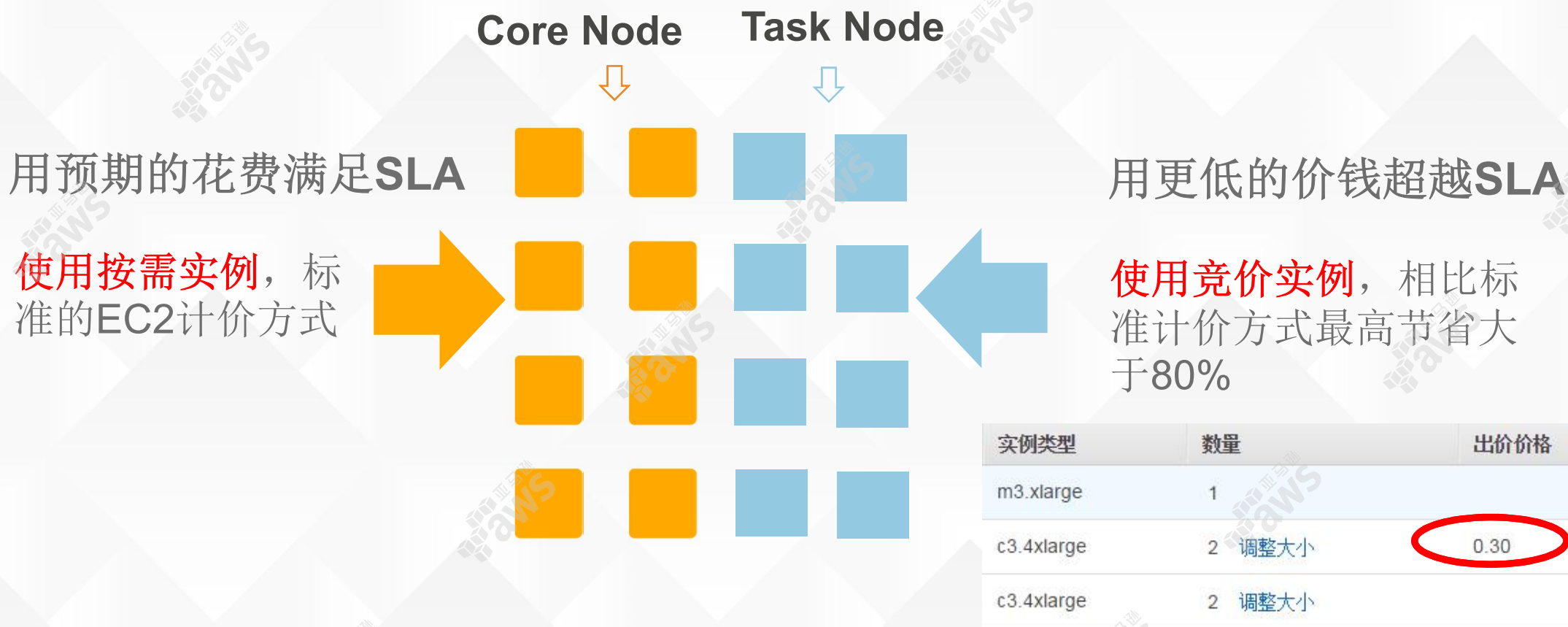
轻松的增加或者减少集群的容量，匹配计算需求。



类型	实例类型	数量
MASTER	m3.xlarge	1
TASK	c3.4xlarge	2 调整大小
CORE	c3.4xlarge	<input type="text" value="2"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

节约成本

充分利用竞价实例



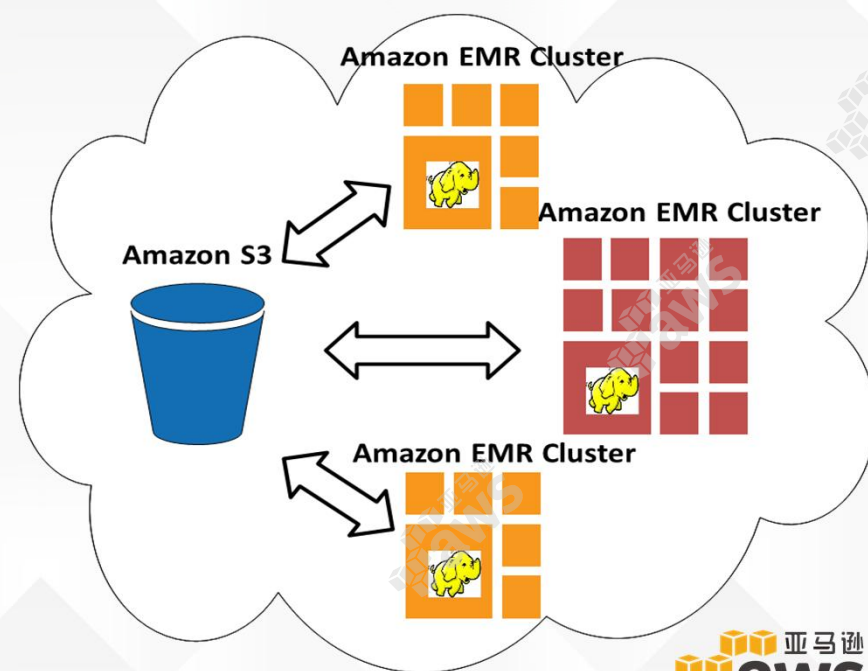


利用Amazon S3作为EMR的存储



使用Amazon S3做为数据持久存储

- Amazon S3
 - 计算和存储分离
 - 99.999999999% 的数据持久性
- 调整EMR集群的大小或者关闭集群的时候没有数据丢失
- 数据集中存储，供多个集群进行分析
- 更容易在集群中引入新技术



EMRFS使得S3的使用更加容易

- Read-after-write一致性
- 使用EMRFS Metadata实现更快的Listing.
- 支持S3的加密
- 对应用层透明 – 只需要读/写到 “s3://...”

从HDFS到Amazon S3

```
hive> create external table temp_user(  
    firstname VARCHAR(64),  
    lastname  VARCHAR(64),  
    address   STRING,  
    country   VARCHAR(64),  
    city      VARCHAR(64),  
    state     VARCHAR(64),  
    web       STRING  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n'  
STORED AS TEXTFILE
```

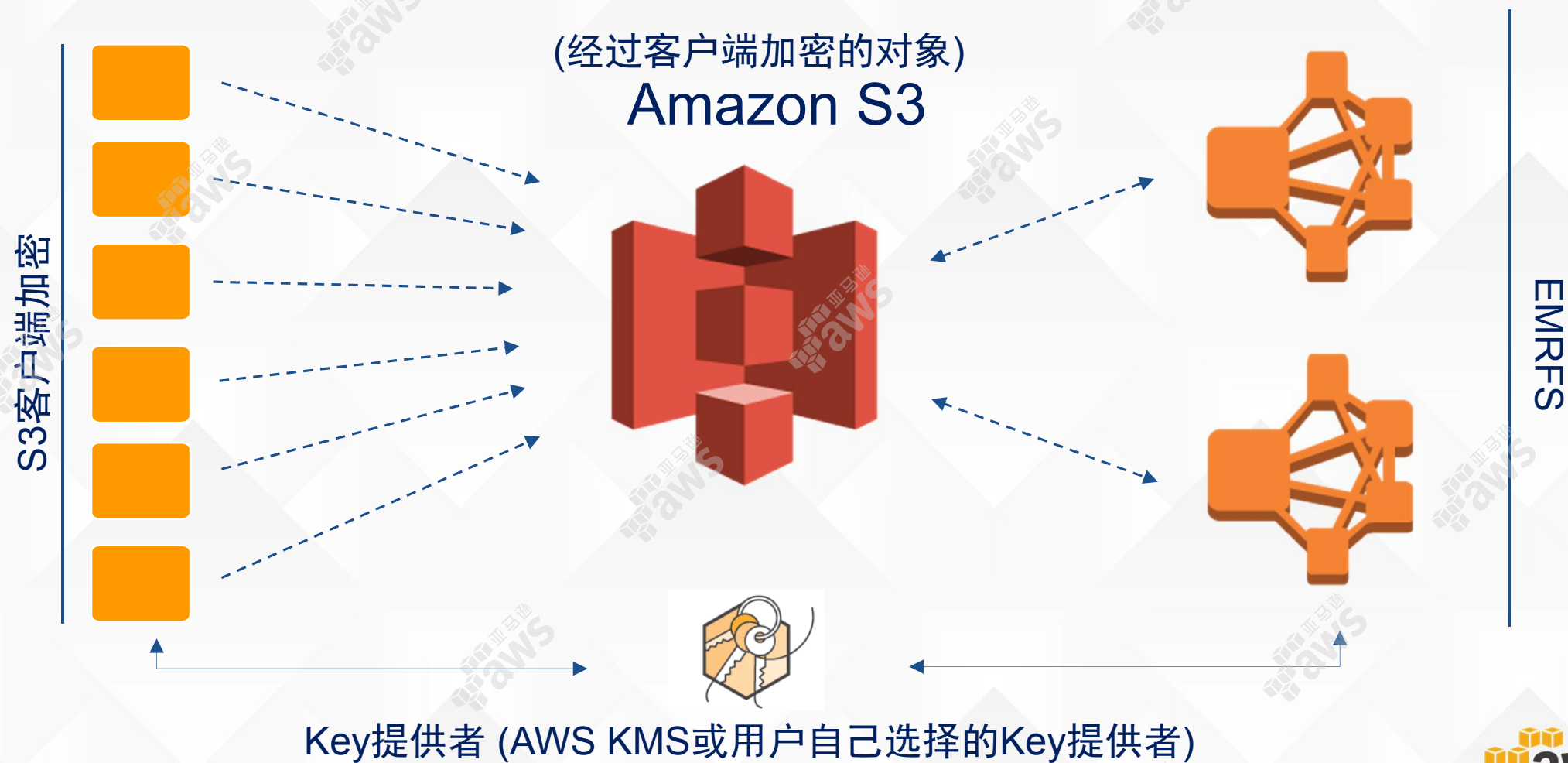
```
LOCATION 'sampledata/userrecord/';
```

从HDFS到Amazon S3

```
hive> create external table temp_user(  
    firstname VARCHAR(64),  
    lastname  VARCHAR(64),  
    address   STRING,  
    country   VARCHAR(64),  
    city      VARCHAR(64),  
    state     VARCHAR(64),  
    web       STRING  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n'  
STORED AS TEXTFILE
```

```
LOCATION 's3://hxyhivetest/userrecord/';
```


EMRFS支持S3加密



使用EMRFS Metadata保证一致性视图且快速的List s3中的对象

- List以及Read-after-write一致性
- 快速的list



List的文件数量	无一致性视图时的时间(秒)	有一致性视图时的时间(秒)
100万	147.72	29.70
10万	12.70	3.69

*测试使用单节点集群实例，类型是m3.xlarge.

HDFS仍然可用

- 迭代型的作业

- 对一份数据进行重复的处理
- 或者考虑使用Spark&RDD

- I/O密集型的作业

- 数据永久存储在S3，用S3DistCp将数据拷贝到HDFS做处理



Amazon EMR上文件的使用



文件格式

- 行式

- Text文件
- Sequence文件
 - 可写对象
- Avro文件
 - 利用Schema描述

- 列式

- Object Record Columnar (ORC)
- Parquet



表逻辑

行式



列式



文件大小

- 避免小文件
 - 避免小于100 MB的文件
 - 每一个mapper处理一个文件,并对应一个JVM,会产生很大开销
- 少量文件, 文件大小接近块的大小
 - 更少的Amazon S3调用开销
 - 更少的Network/HDFS开销

如何处理小文件

- 使用S3DistCP将多个小文件组合起来
 - S3DistCP根据模式以及目标路径将小文件组合成大文件
 - 提供目标文件的大小以及压缩的方式

文件压缩

- 压缩S3上存储的数据文件
 - 减少S3和EMR之间的网络流量
 - 增加作业完成的速度
- 压缩Mapper和Reducer的输出文件

在Hadoop1, Amazon EMR使用LZO 压缩节点之间传递的数据, 在Hadoop2,使用Snappy

选择正确的压缩方式

- 对于时间敏感的任务，选择压缩速度快的方式
- 对于空间敏感的任务，选择压缩率高的方式
- 对于混合型的任务，可以选择gzip

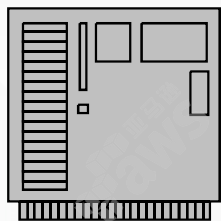
压缩算法	是否可切割	压缩率	压缩和解压缩速度
gzip	否	高	中
bzip2	是	很高	慢
LZO	是	低	快
Snappy	否	低	很快



Amazon EMR的设计模式



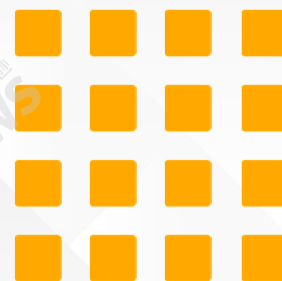
例一：批处理



每小时有GB的日志数据写到S3



存储输入/输出数据



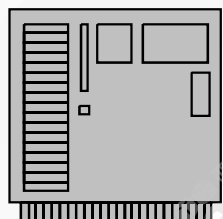
每天启动EMR集群,使用Hive处理数据



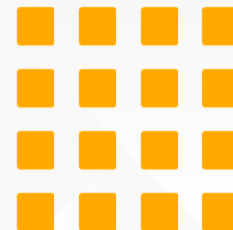
每天250个Amazon EMR作业, 处理30TB数据

<http://aws.amazon.com/solutions/case-studies/yelp/>

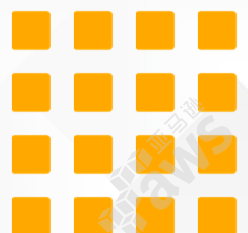
例二：长期运行的集群



数据写入S3



每天启动集群做ETL，
数据写入HBase



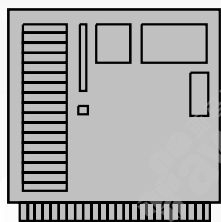
APACHE
HBASE

7x24小时运行，供业务人员查询
数据在S3中做备份



数据展现

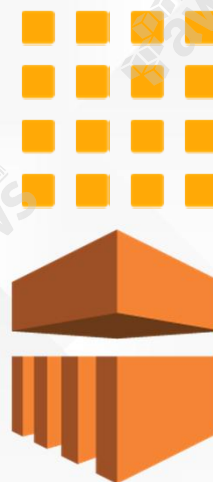
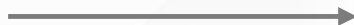
例三：交互式查询



每天TB级的日志数据



日志数据存储在S3

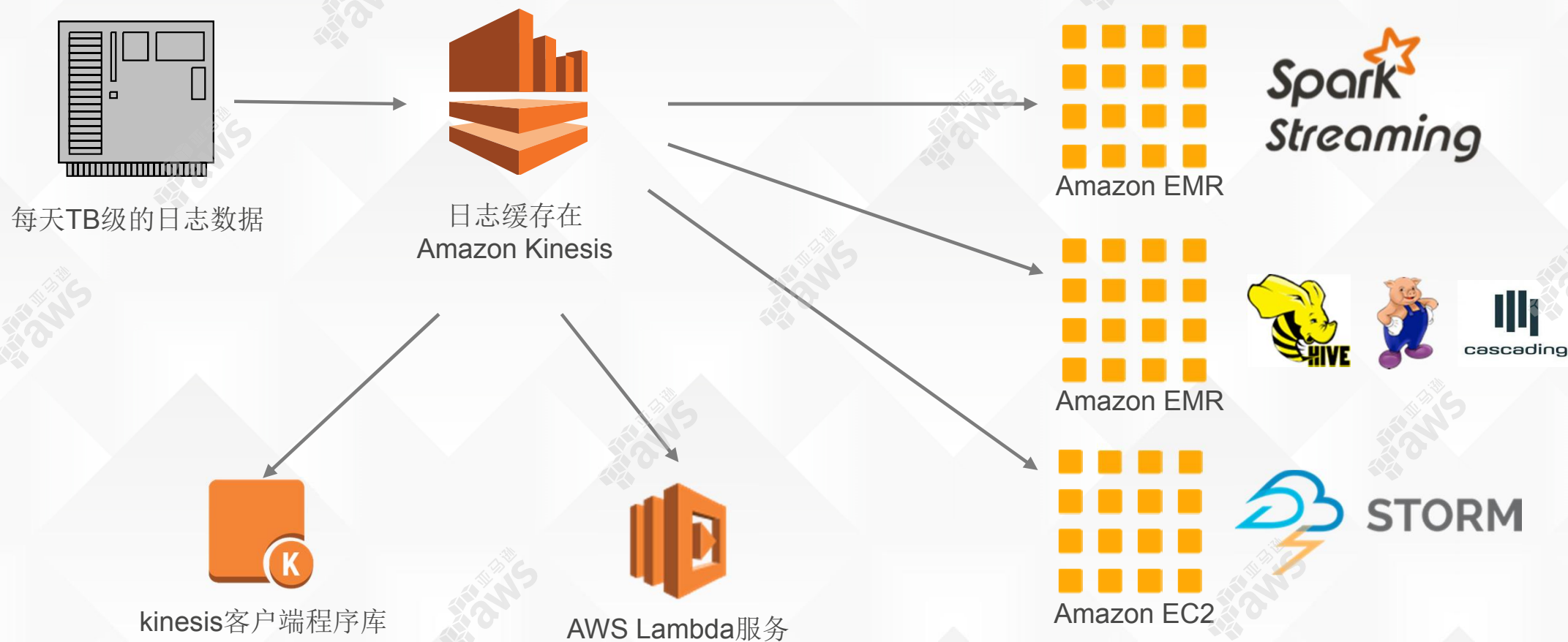


NETFLIX

使用Presto进行PB级数据查询

<http://nflx.it/1dO7Pnt>

例四：流式数据处理



要点

- 使用AWS控制台、命令行、SDK快速搭建集群。使用BootStrap功能安装其他应用。
- 使用S3作为永久存储，使用创建Presto、Hive、Spark等集群进行查询
- 按需使用，不用的时候关闭集群。
- EC2竞价实例用于可随时停止的任务，EC2预留实例用于长期运行的任务。
- 使用Cloud Watch监控集群资源使用情况，例如当超过1小时的时间没有mapper运行的时候，关闭部分实例。

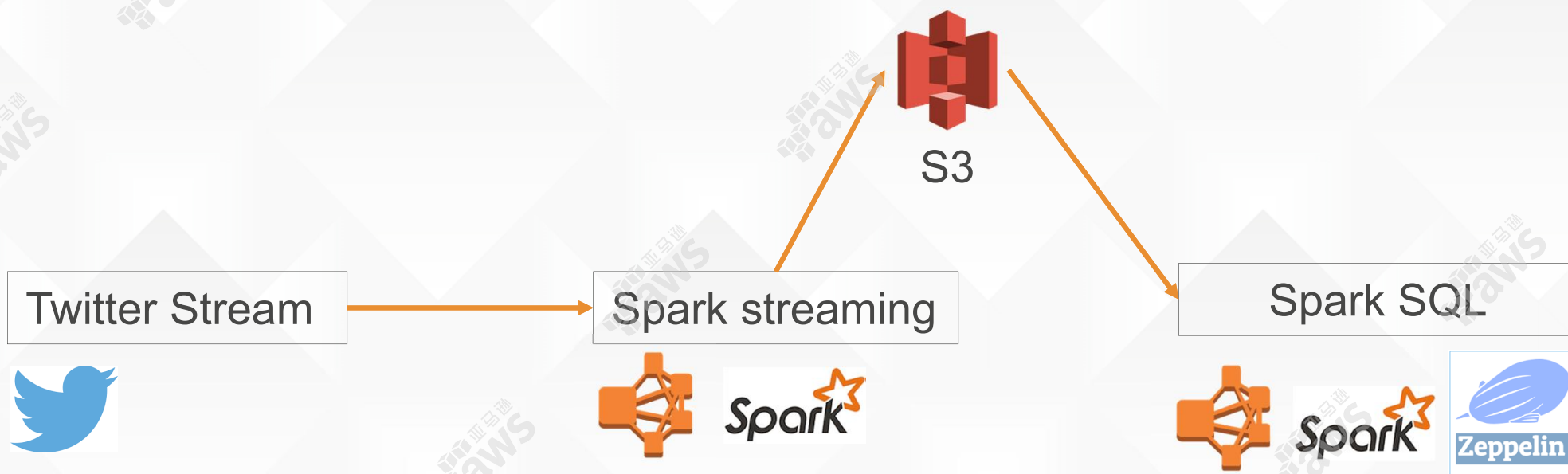


现场演示



演示：利用Spark读取Twitter流，并展现

- 快速部署
- 改变集群大小
- 竞价实例
- 用S3做存储
- 数据展现



```
1. aws emr create-cluster
2. --region us-west-2 \
3. --name "Tweetalizer" \
4. --release-label emr-4.1.0 \
5. --applications 'Name=Zeppelin-Sandbox'
6. --ec2-attributes KeyName=bjskey, AvailabilityZone=us-west-2
7. --log-uri s3://spark/log \
8. --instance-groups \
9.   Name=Master,InstanceGroupType=MASTER,InstanceType=m3.xlarge,InstanceCount=1 \
10.  Name=Core,InstanceGroupType=CORE,InstanceType=r3.2xlarge,InstanceCount=3 \
11.  Name=Task,InstanceGroupType=TASK,InstanceType=r3.2xlarge,InstanceCount=3,\
12.    BidPrice=0.3 \
13. --Steps
    Name=S3Copy,Jar=s3://elasticmapreduce/libs/script-runner/script-
runner.jar,Args=["/usr/bin/aws,s3,cp","s3://spark/xx.jar",/home/hadoop/lib/jarfile] \

    Type="Spark",
    Name="tweetalizer",ActionOnFailure=CONTINUE,
    Args=["--master","yarn","--class","org.spark.PopularHashTagsCounter","/home/hadoop/lib/"]
```



```
8
9 public class CreateCluster {
10
11     public static void main(String[] arg)
12     {
13         //创建一个emr_client对象.
14         AWSCredentials credentials = new ProfileCredentialsProvider().getCredentials();
15         AmazonElasticMapReduceClient emr = new AmazonElasticMapReduceClient(credentials);
16         emr.setEndpoint("elasticmapreduce.us-west-2.amazonaws.com");
17
18         StepFactory stepFactory = new StepFactory();
19
20         //创建debug step
21         StepConfig enableddebugging = new StepConfig()
22             .withName("Enable debugging")
23             .withActionOnFailure("TERMINATE_JOB_FLOW")
24             .withHadoopJarStep(stepFactory.newEnableDebuggingStep());
25
26         //创建从S3中拷贝jar包的step
27         StepConfig s3copy = new StepConfig()
28             .withName("s3copy")
29             .withHadoopJarStep(new HadoopJarStepConfig()
30                 .withJar("s3://emrdevelop/command-runner-0.1.0.jar")
31                 .withArgs("aws", "s3", "cp", "s3://emrdevelop/tweetalyzer-assembly-0.1.0.jar", "/home/hadoop/lib/")
32                 );
33
34         //创建spark的step
35         StepConfig tweetalyzer = new StepConfig()
36             .withName("tweetalyzer")
37             .withHadoopJarStep(new HadoopJarStepConfig()
38                 .withJar("s3://emrdevelop/command-runner-0.1.0.jar")
39                 .withArgs("spark-submit", "--master", "yarn", "--class", "org.zezutom.spark.tweetalyzer.PopularHashTagsCounter", "/h
40
41
```

```

42
43 //配置master, core, task
44 InstanceGroupConfig master = new InstanceGroupConfig()
45     .withInstanceCount(1)
46     .withInstanceRole("MASTER")
47     .withInstanceType("m3.xlarge");
48 InstanceGroupConfig core = new InstanceGroupConfig()
49     .withInstanceCount(1)
50     .withInstanceRole("CORE")
51     .withInstanceType("c3.4xlarge");
52 InstanceGroupConfig task = new InstanceGroupConfig()
53     .withInstanceCount(1)
54     .withInstanceRole("TASK")
55     .withInstanceType("c3.4xlarge")
56     .withMarket("SPOT")
57     .withBidPrice("0.3");
58
59 //集群配置
60 RunJobFlowRequest request = new RunJobFlowRequest()
61     .withName("Tweetalizer")
62     .withReleaseLabel("emr-4.1.0")
63     .withApplications(new Application().withName("Zeppelin-Sandbox"))
64     .withSteps(enableddebugging, s3copy, tweetalizer)
65     .withLogUri("s3://emrdevelop/log")
66     .withJobFlowRole("EMR_EC2_DefaultRole")
67     .withServiceRole("EMR_DefaultRole")
68     .withInstances(new JobFlowInstancesConfig()
69         .withEc2KeyName("OregonKey")
70         .withInstanceGroups(master, core, task)
71     );
72
73 //运行集群
74 RunJobFlowResult result = emr.runJobFlow(request);
75 System.out.println(result);
76 }
77 }

```

用Scala写Spark程序

Spark Streaming中写数据：

```
rdd.saveAsTextFile("s3://sparkdevtest/output")
```

Spark Sql中读数据：

```
val lines = sc.textFile("s3://sparkdevtest/output")
```



有奖问答



1

问：列举至少3个Amazon EMR中提供的开源服务

答：Hadoop Mapreduce, HDFS, Spark, Hive, pig, Hue, Zeppelin, Impala, Presto, Oozie等

2



问：EMR使用的AWS的存储服务是什么，并列举出至少1个使用这种存储的优势。

答：S3.

计算和存储解耦, 节约资源。

便于同时服务多个集群。

数据持久性高(11个9)。

3



问：操作Amazon EMR除了从控制台进行，还有什么方式？至少说出一种

答：
命令行
SDK



Thank You

