IBM

Insight2014

The Conference for Big Data and Analytics

Data Science Using Big R for In-Hadoop Analytics

Brandon MacKenzie, IBM Software Group

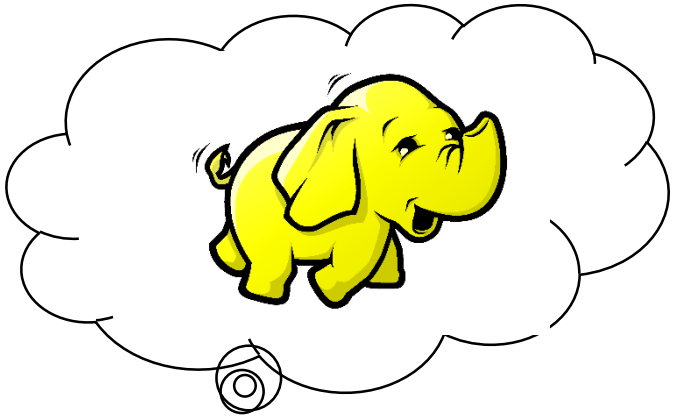
Meetup®

SEIZE THIS MOMENT

#ibminsight

Meetup®

Working together to build a Smarter Planet



- Rafael Coss
- WW Big Data Technical Enablement Mgr
- rcoss@us.ibm.com
- @racoss

Big Data Developer Meetups @ Vegas

Where: Delano Las Vegas Hotel
[Attached to Mandalay Bay & Resort]
Room: Sienna C

Sunday, October 26 2:00pm - 4:00pm
Data Science using Big R for in-Hadoop Analytics

Monday, October 27 3:00pm - 5:00pm
Big SQL 3.0: SQL on Hadoop without Compromise

Tuesday, October 28 3:00pm - 5:00pm
The Internet of Things & Geospatial Analytics

Wednesday, October 29 4:00pm - 5:00pm
Getting started with Hadoop in the Cloud

Find your local Big Data Meetup
bigdatadevelopers.meetup.com

Big Data for Social Good Challenge

Powered by IBM + Hadoop

- Big Data, Big Issues, Big Challenges
  - Let's build a smarter planet together with Hadoop
- Hosted by Challenge Post
  - ibmhadoop.challengepost.com
- When?
  - Nov 10, 2014 – Mar 3, 2
- Prizes?
  - $40,000

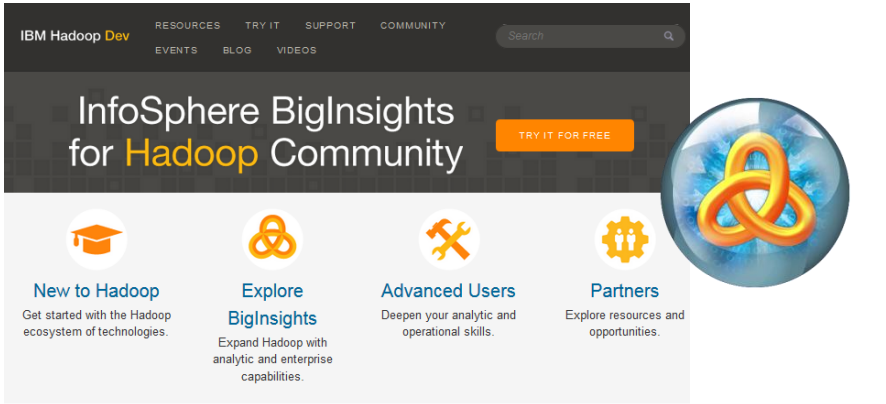
Want to learn more?

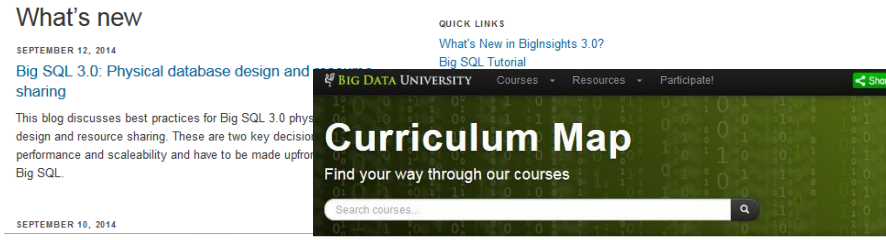
- **Get it**
  - BigInsights Quick Start Edition VM
    - ibm.co/quickstart
  - Analytics for Hadoop Service
    - bluemix.net
  - Big SQL Tech Preview
    - bigsql.imdemocloud.com

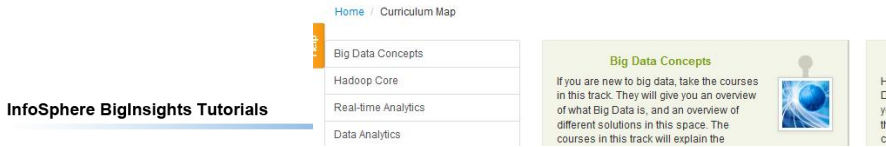
IBM Analytics for Hadoop

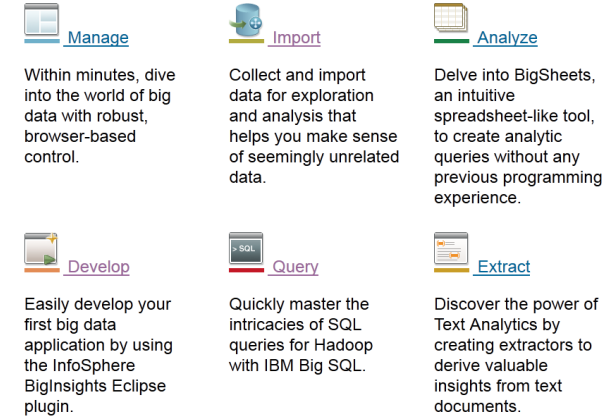
- **Learn it**
  - Follow online tutorials
    - ibm.biz/tutorial
  - Enroll in online classes
    - BigDataUniversity.com

- Hadoop Dev
  - Links all available
  - Watch video demos, read articles, etc.
  - https://developer.ibm.com/hadoop







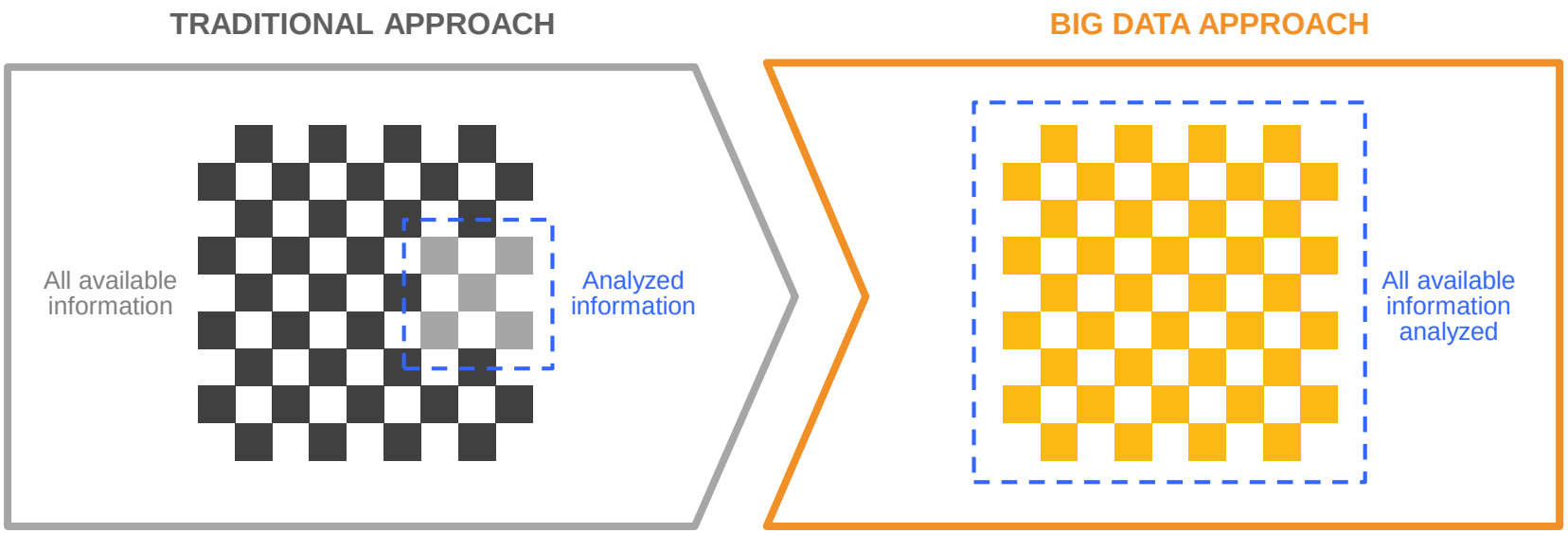


Please Note

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Leverage more of the data being captured



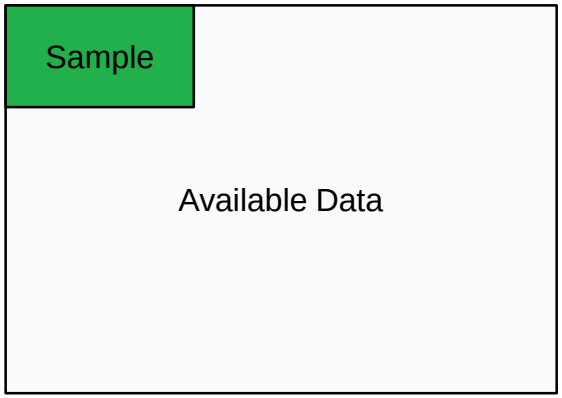
Analyze small subsets of information

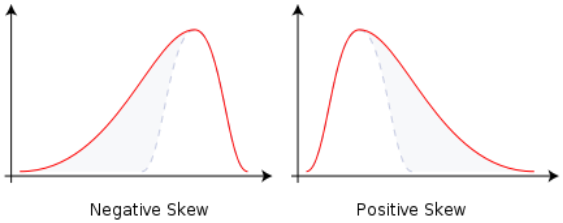
Analyze ***all*** information

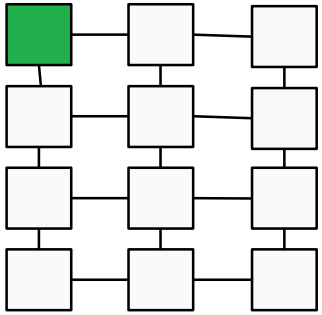
Challenges in Expressing and Running Big Analytics

```
package gnmf;

import java.io.IOException;
import java.net.URISyntaxException;

import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapred.JobConf;


public class MatrixGNMF
{
  public static void main(String[] args) throws IOException, URISyntaxException
  {
    if(args.length < 10)
    {
      System.out.println("missing parameters");
      System.out.println("expected parameters: [directory of v] [directory of w] [directory of h] " +
          "[k] [num mappers] [num reducers] [replication] [working directory] " +
          "[final directory of w] [final directory of h]");
      System.exit(1);
    }

    String vDir      = args[0];
    String wDir      = args[1];
    String hDir      = args[2];
    int k          = Integer.parseInt(args[3]);
    int numMappers    = Integer.parseInt(args[4]);
    int numReducers    = Integer.parseInt(args[5]);
    int replication    = Integer.parseInt(args[6]);
    String outputDir  = args[7];
    String w
    String h

    JobConf
    String v
    String w
    String h

    FileSyst


    vDirecto
    hDirecto
    wDirecto

    String workingDirectory;
    String resultDirectoryX;
    String resultDirectoryY;

    long start = System.currentTimeMillis();
    System.gc();
    System.out.println("starting calculation");
    System.out.print("calculating X = WT * V... ");
    workingDirectory = UpdateWHStep1.runJob(numMappers, numReducers, replication,
        UpdateWHStep1.UPDATE_TYPE_H, vDirectory, wDirectory, outputDir, k);
    resultDirectoryX = UpdateWHStep2.runJob(numMappers, numReducers, replication,
        workingDirectory, outputDir);
    FileSystem.get(mainJob).delete(new Path(workingDirectory));

    System.out.println("done");
    System.out.print("calculating Y = WT * W * H... ");
    workingDirectory = UpdateWHStep3.runJob(numMappers, numReducers, replication,
        wDirectory, outputDir);
    resultDirectoryY = UpdateWHStep4.runJob(numMappers, replication, workingDirectory,
        UpdateWHStep4.UPDATE_TYPE_H, hDirectory, outputDir);
    FileSystem.get(mainJob).delete(new Path(workingDirectory));
    System.out.println("done");

    System.out.print("calculating H = H .* X ./ Y... ");
    workingDirectory = UpdateWHStep5.runJob(numMappers, numReducers, replication,
        hDirectory, resultDirectoryX, resultDirectoryY, hFinalDir, k);
    System.out.println("done");
```

Java MapReduce Implementation (>1500 lines of code)







Productivity
- Data scientists
- Explicitly code parallelism
- Hand-crafted specific platforms

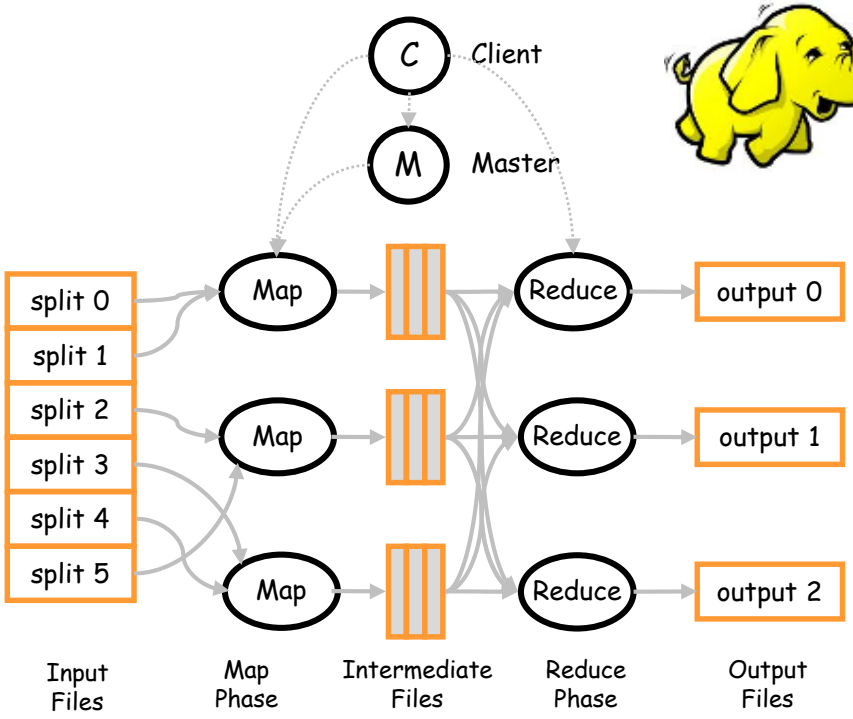
Big Data
- Sampling, 1% vs 100%
- Skewed, Sparse
- Numerical Accuracy

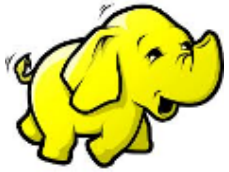
Scalability and Optimizations
- Clusters 1000s of machines
- Fixed execution plans
- Different datasets

Open Source R's Strengths… and Weaknesses

Free	Data Exploration	Large Data Volumes
Vibrant Growing Community	Descriptive Statistics and Machine Learning	Scalable Algorithms
Industry Commitment	Visualization	Not Naturally Hadoop Friendly

Hadoop gets much of the attention for new workloads





- Pioneered by Google and Yahoo!
- Framework for writing applications to process vast data sets
- Minimize data movement
- More cost efficient than traditional data warehouses for select problems
- Rapidly evolving ecosystem
- New frameworks building on Hadoop innovation – Spark

100% Open Source Hadoop, but Enterprise Grade

Value-Added Capabilities

SQL on Hadoop
Big SQL – optimized ANSI compliant SQL

Application Tooling
Toolkits and accelerators

Search
BigIndex and Data Explorer

Data Exploration
BigSheets "schema-on-read"

Predictive Modeling
Big R – scalable data mining

Text Analytics
Advanced text processing with AQL

Real-time Analytics
InfoSphere Streams

Data Governance and Security
Data Click, LDAP, Secure cluster

Storage Integration
GPFS - POSIX Distributed Filesystem

Enterprise Features
Adaptive MapReduce, Recoverable jobs

100% Standard Apache Open-Source Components

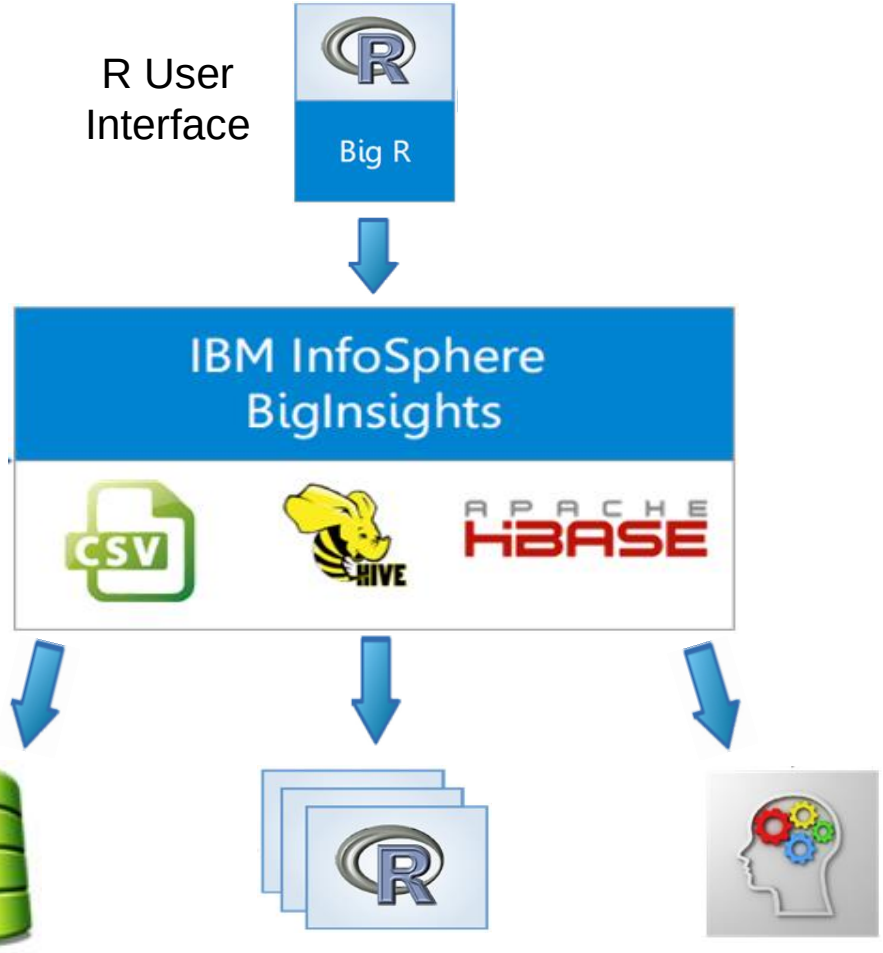
Oozie	Jaql	Zookeeper	Hive	HCatalog
HDFS	MapReduce	HBase	Flume	Sqoop
YARN*	Spark*	Avro	Pig	Solr/Lucene

3 Key Capabilities in Big R

End-to-end integration of R into InfoSphere BigInsights

1. **Use of R as a language on big data**
   - Scalable data processing
2. **Running <u>native R</u> functions in Hadoop**
   - Can leverage existing R assets (code and CRAN packages)
3. **Running scalable algorithms <u>beyond R</u> in Hadoop**
   - Wide class of algorithms and growing
   - R-like syntax to develop new algorithms and customize existing algorithms

Big R Architecture



1 Scalable Data Processing

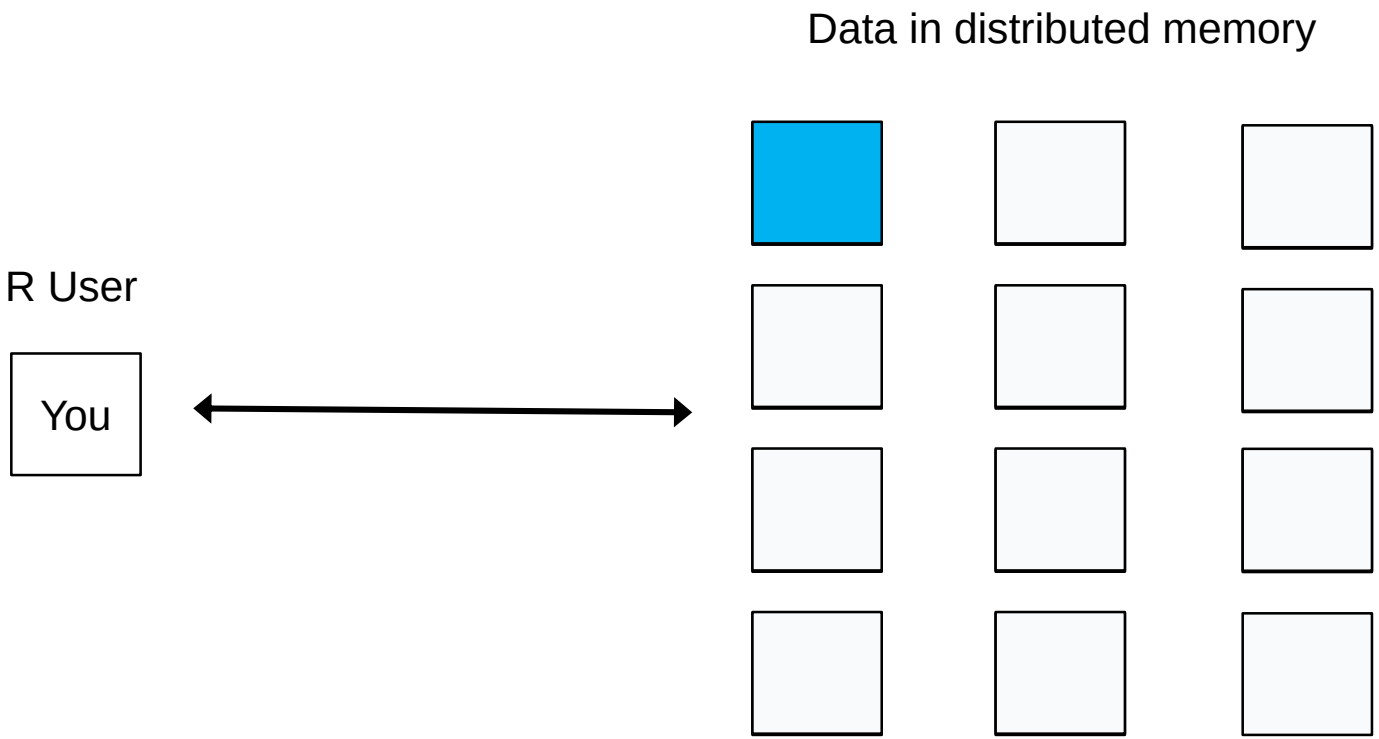
2 Native R functions

3 Scalable Algorithms

Big R Classes

Big R Classes	Inspired by R	Functionality
bigr.frame	data.frame	A proxy to an arbitrarily larger tabular dataset.
bigr.vector	vector	A proxy to an arbitrarily large uni-dimensional dataset
bigr.list	list	A set of arbitrary objects as a result of partitioned execution.
bigr.matrix	matrix	A proxy to an arbitrarily large numeric dataset persisted in HDFS.

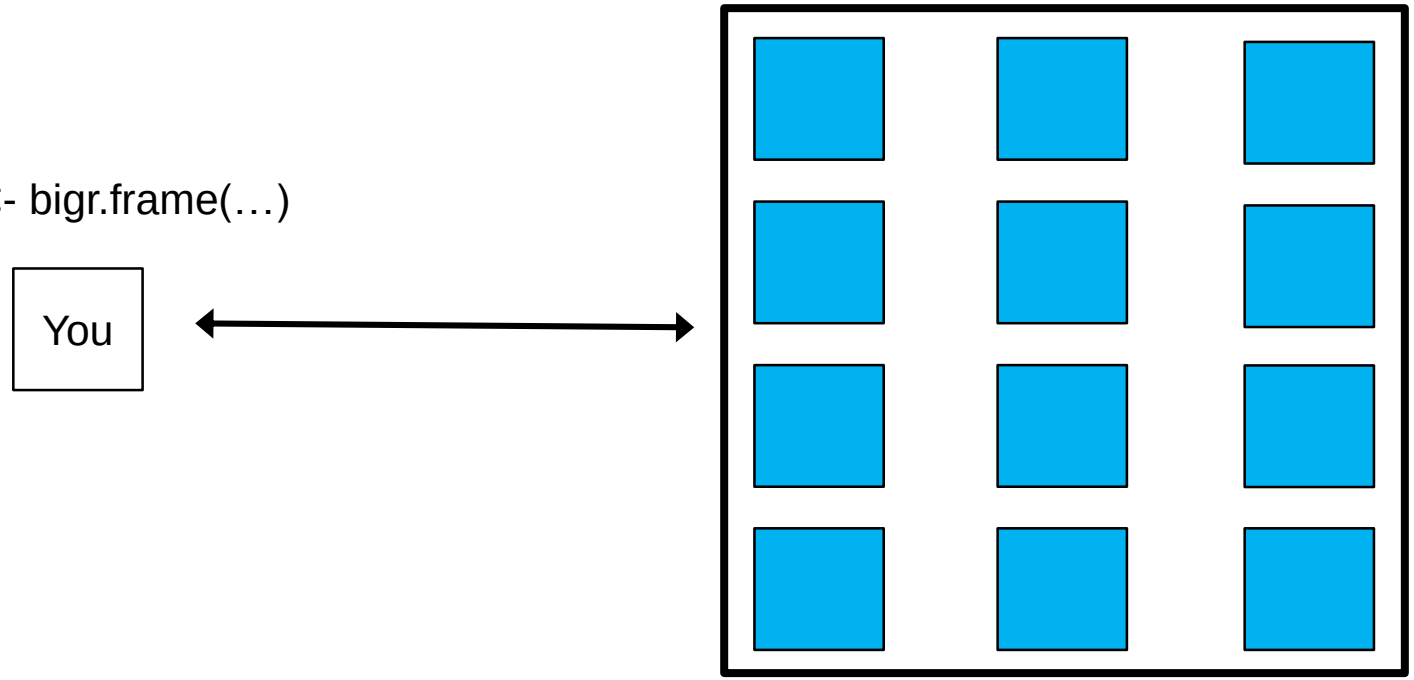
Data in Hadoop: Open Source R on a Single Node



Big R Data Structures: Proxy to Entire Dataset

Appears and acts like all of the data is on your laptop

data <- bigr.frame(…)



Big R Operators

Big R Operators	Type	Applicable to
+ - * / %% %/% ^	Arithmetic	bigr.vectors and R data types
& | !	Logical	bigr.vectors and R data types
== > < != <= >= %in%	Relational	bigr.vectors and R data types
[] [,] $ $<-	Transformation	bigr.vectors, bigr.frames, and bigr.lists

Big R Query Functions

Functions	Description
head(), tail()	First or last k elements of a bigr.frame or bigr.vector.
str(), show(), print()	Visualize a bigr.frame, bigr.vector, bigr.list.
colnames(), coltypes()	Assign column names and types for a bigr.frame.
attach(), with()	Direct access to the column of a bigr.frame.
sort(), merge()	Sort a bigr.frame or bigr.vector. Join two bigr.frames.
bigr.persist()	Export a bigr.frame, bigr.vector, or bigr.list to persistent data source.
ifelse()	Recode a bigr.vector.

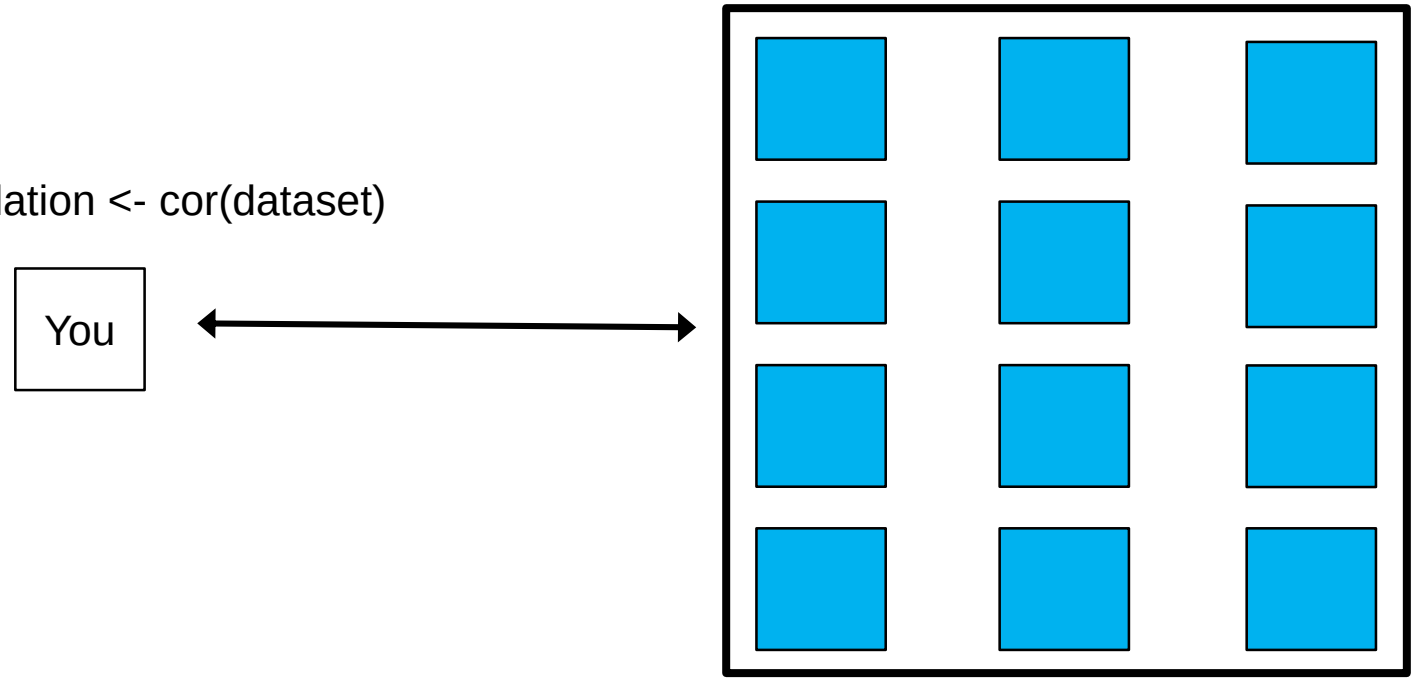
Big R Analytics Functions

Functions	Description
mean(), sd(), var(), cov(), cor(), quartiles()	Univariate / bivariate statistics.
summary(), min(), max(), range(), sum(), count(), mean()	Aggregate functions. Could be applied on the entire data or on a group basis via summary().
unique(), table()	Distinct values and counts for each value.
abs(), sign(), log(), pow(), sqrt()	Arithmetic functions.

Out-of-box Big R Functions: Seamlessly Compiled Jobs

MapReduce job runs over the entire dataset

dataCorrelation <- cor(dataset)



Machine Learning with Big R (Beta)

Big R leverages **SystemML**, declarative distributed machine learning engine

From IBM Research (5+ years of development)

Features

- Compiler automatically parallelizes and optimizes for performance
  - Based on data characteristics and Hadoop configuration
- Offers a handful of scalable algorithms/functions out-of-the-box:
  - ✓ Data Preparation
  - ✓ Descriptive Statistics
  - ✓ Machine Learning Algorithms

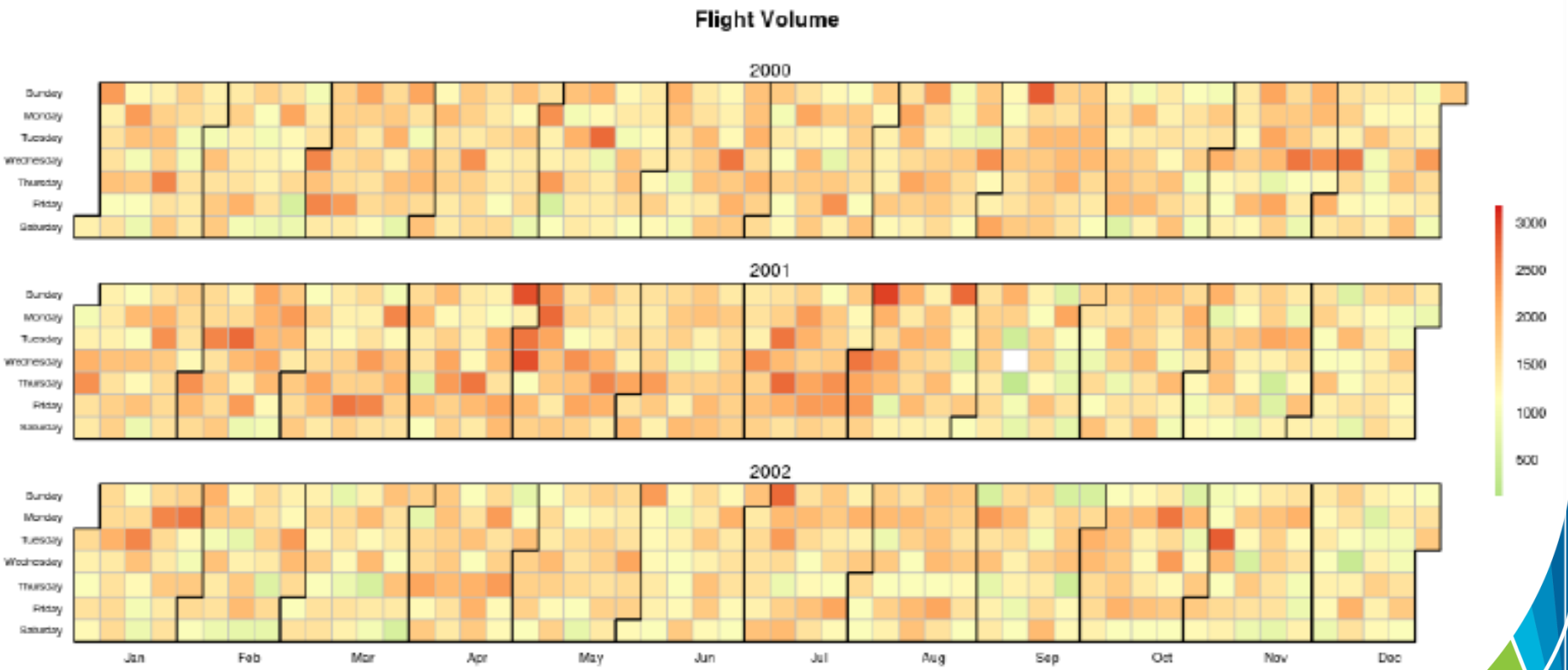
Example 1: Using R as a Query Language

```
# Connect to BigInsights
> bigr.connect(host="192.168.153.219", port=7052, user="biadmin", password="xyz")

# Construct a bigr.frame to access large data set
> air <- bigr.frame(dataSource="DEL", dataPath="airline_demo.csv", …)

# Filter flights delayed by 15+ mins at departure or arrival
> airSubset <- air[air$Cancelled == 0
                 & (air$DepDelay >= 15 | air$ArrDelay >= 15),
               c("UniqueCarrier", "Origin", "Dest",
                 "DepDelay", "ArrDelay", "CRSElapsedTime")]


# What percentage of flights were delayed overall?
> nrow(airSubset) / nrow(air)
[1] 0.2269586

# What are the longest flights?
> bf <- sort(air, by = air$Distance, decreasing = T)
> bf <- bf[,c("Origin", "Dest", "Distance")]

> head(bf, 3)
Origin Dest Distance
1    HNL  JFK     4983
2    EWR  HNL     4962
3    HNL  EWR     4962
```

Example 2: Visualizations with Big R

Example 1: Display the flight volume in the early 2000's

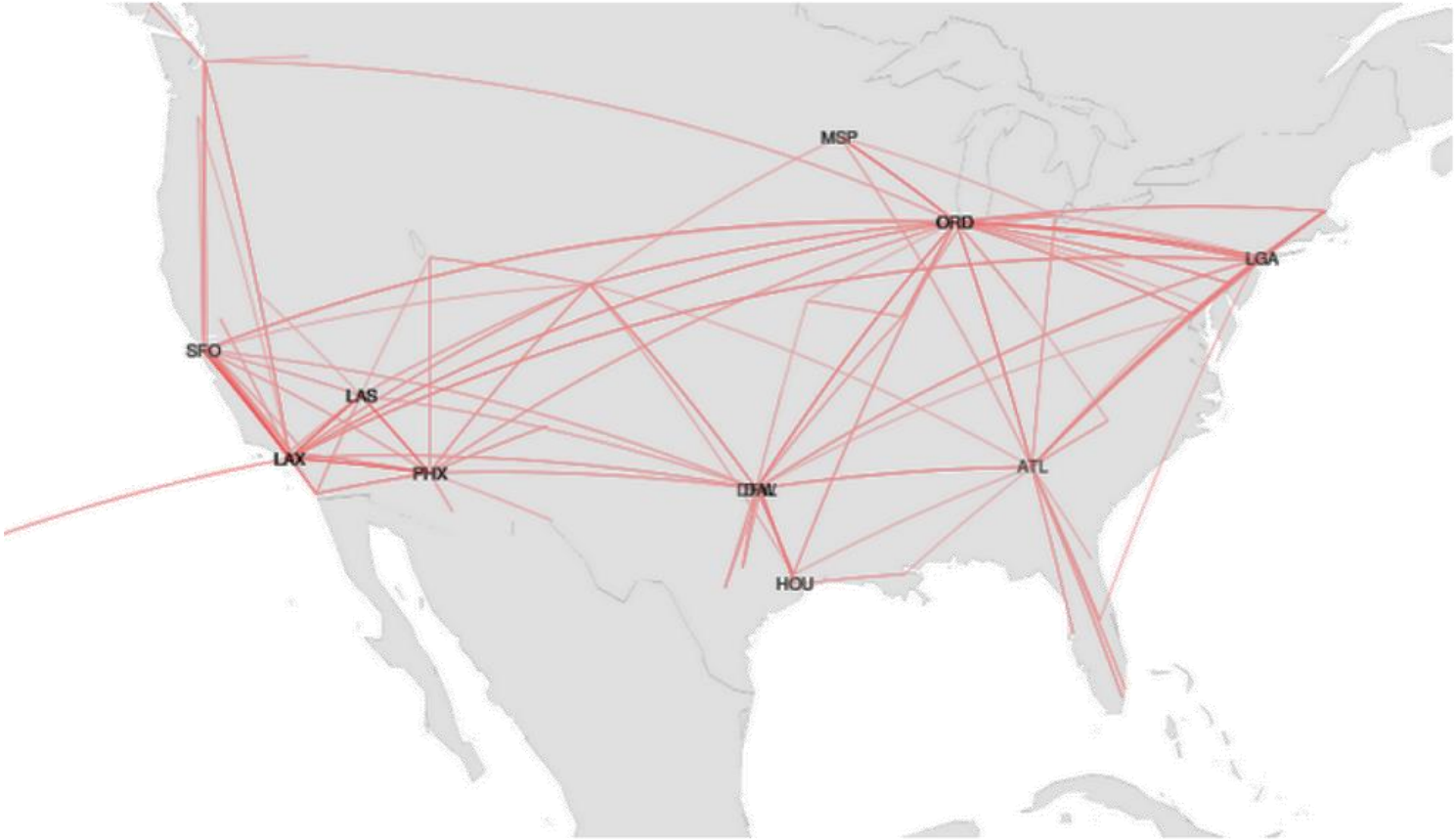
```
> summary(count(.) ~ Month + DayofMonth + Year,
    data=air[air$Year %in% c(2000, 2001, 2002) &
      air$Canceled == 0, ])
```



Example 3: Visualizations with Big R

Example 2: Illustrate the busiest flight routes in the US

```
summary(count(.) ~ Origin + Dest, data=airline)
```



Partitioned Execution

Virtually any R function can run on the cluster:

- Data are partitioned in R-able chunks
- Big R spawns R instances on each node
- R executes the given function on each partition

Follows R's **<u>*apply() paradigm</u>**

- Like R's tapply(), Big R has groupApply()

Partition criteria:

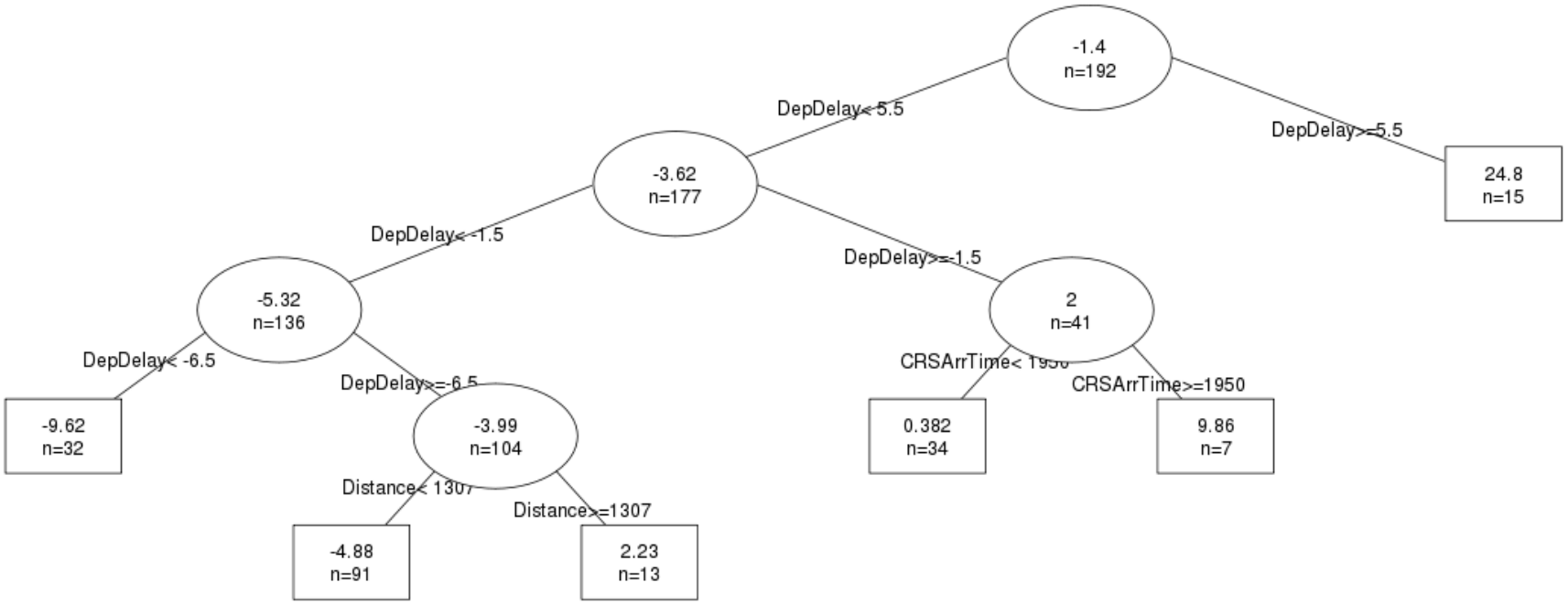
- One or more grouping columns, random numbers, calculated columns, or a fixed number of rows

Example 4: Model Building with Partitioned Execution

```
# Filter the airline data on United and Hawaiian
bf <- air[air$UniqueCarrier %in% c("HA", "UA"),]

split <- bigr.sample(bf, perc=c(0.7, 0.3)
train <- split[[1]]
test <- split[[2]]


buildModel <- function(df) {
    library(rpart)
    predcols <- c('ArrDelay', 'DepDelay', 'DepTime', ‘CRSArrTime’, 'Distance')
    model <- rpart(ArrDelay ~ ., df[,predcols])
    return (model)
})


# Build one regression-tree model per airline
models <- groupApply(data = train,
                     groupingColumns = train$UniqueCarrier,
                     rfunction = buildModel))

# Pull all models to client
rmodels <- bigr.pull(models)
```

Runs as-is on cluster
<u>on each of the groups</u>

Example 4: Building Models with Partitioned Execution



IBM InfoSphere BigInsights – Text Analytics

Improve time to value

- Distills unstructured text into structured information
  - Sentiment analysis
  - Consumer behavior
  - Illegal or suspicious activities
- Parses text and interprets meaning with annotators
- Understands the context in which the text is analyzed
- Features pre-built extractors for names, addresses, phone numbers and more
  - Built-in support for English, Spanish, French, German, Portuguese, Dutch, Japanese, Chinese

IBM InfoSphere BigInsights – Text Analytics

Extract information from unstructured sources for business insight

Customer: I'm calling because I received an incorrect bill. I just paid my bill two days ago, and my payment is not reflected

Agent: Sorry for the inconvenience. May I have your Account Number, please?

Customer: 15635764 – wait – I meant 15365764

Agent: For verification purposes, can I get your name and birth date?

Customer: Marge Simpson, Nov 23, 1975 and the account is under my Husband's name, Homer

Agent: Thank you for that information. Per our system, you did pay your bill Aug. 12th

```
<call _center_record
    trans_id=132436>
    <cust_id>15365764</cust_id>
    <account_holder>
        Homer Simpson
    </account_holder>
    <caller_birthdate>
        1975-11-23
    </caller_birthdate>
    <inquiry>balance</inqury>
    <balance>0</balance>
    <pmt_date>2014-08-12</pmt_date>
    <cred_score>3.9</cred_score>
    ..
    ..
</call_center_record>
```

Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Thank You