Laurence Liew

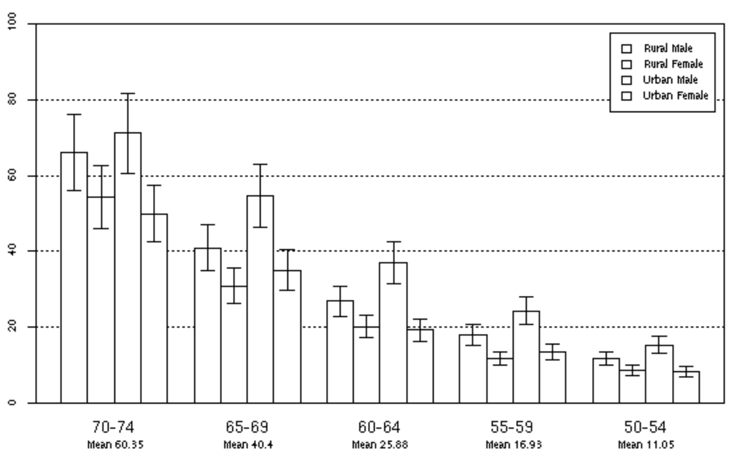
General Manager, APAC

Revolution Analytics

laurence.liew @ revolutionanalytics.com | +65 9029 4312

Big Data Analytics with R

A Hadoop and HPC Cluster Perspective



Laurence Liew

General Manager, APAC

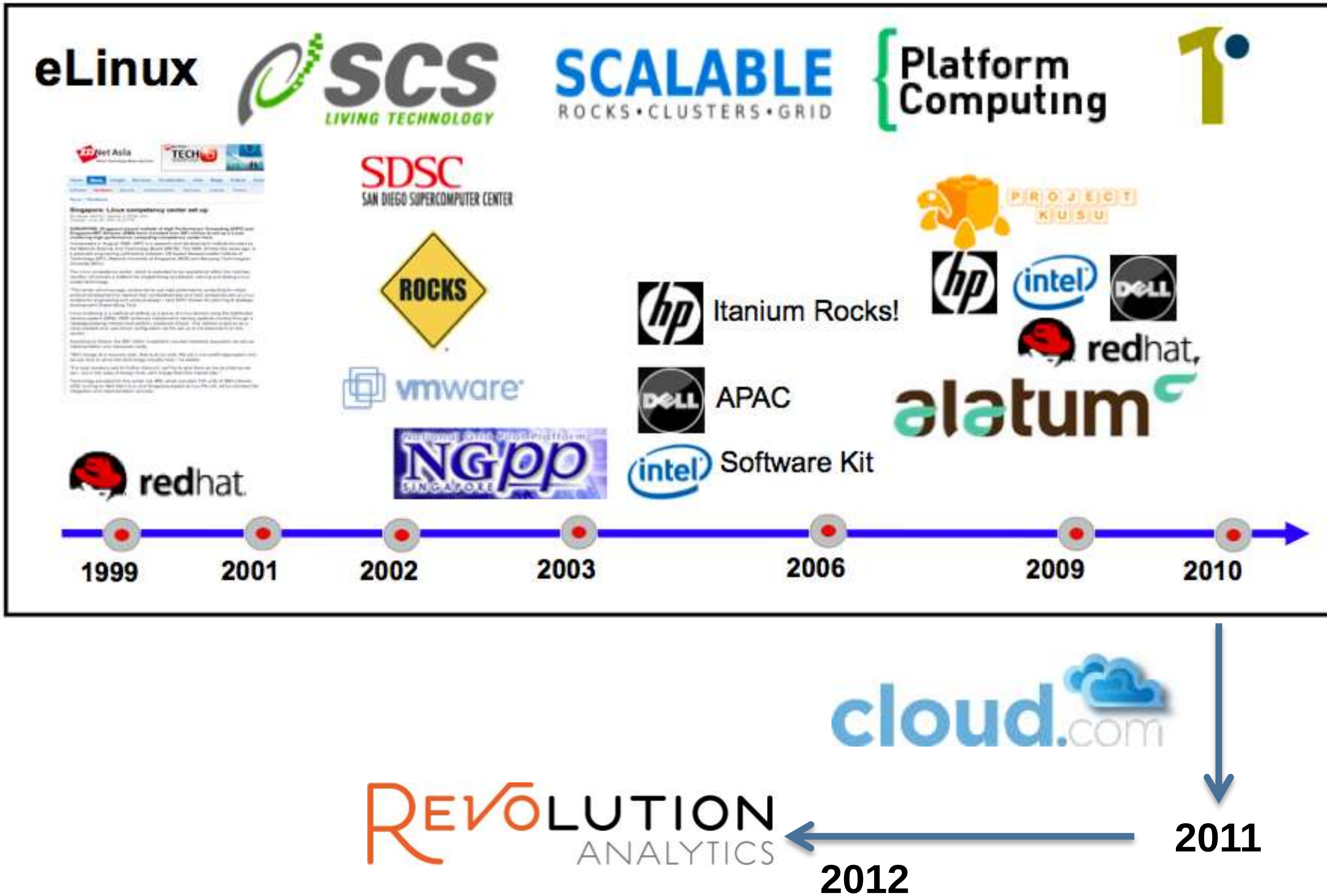
Revolution Analytics

laurence.liew @ revolutionanalytics.com | +65 9029 4312

Agenda

- Introduction
- What is Big Data and Why Big Data?
- Why R?
- High Performance Analytics on Big Data with HPC Clusters and R
- High Performance Analytics on Big Data with Hadoop Clusters and R
- Enterprise Deployment of Big Data Analytics
- Technical Walk-thru and demos

Background


eLinux
SCS
LIVING TECHNOLOGY
SCALABLE
ROCKS•CLUSTERS•GRID
Platform Computing
SDSC
SAN DIEGO SUPERCOMPUTER CENTER
PROJECT KUSU
ROCKS
hp
Itanium Rocks!
intel
DELL
redhat
APAC
alatum
vmware
NGPP
SINGAPORE
Software Kit
redhat
1999
2001
2002
2003
2006
2009
2010
cloud.com
REVOLUTION ANALYTICS
2011
2012

Corporate Overview & Quick Facts

"Revolution Analytics is the leading commercial provider of software and support for the open-source R statistical computing language."

Founded	2008 (as REvolution Computing)	**Number of Employees**	40+
Office Locations	Palo Alto (HQ), Seattle (Eng), Singapore	**Number of customers**	100+
CEO	David Rich	**Investors**	Northbridge Venture Partners, Intel Capital, Presidio Ventures

150+ Corporate Customers and growing

Finance & Insurance

Healthcare & Life Sciences

Academic & Gov't

Consumer & Info Svcs

Manuf & Tech

REVOLUTION ANALYTICS

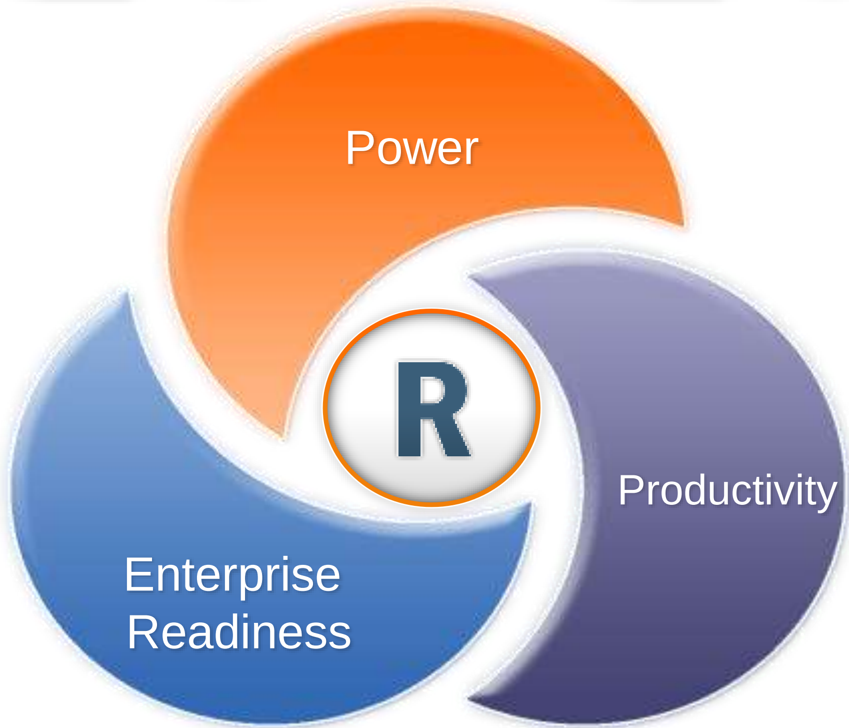
OPEN SOURCE ANALYTICS FOR THE ENTERPRISE

- **Most advanced statistical analysis software available**
- **Half the cost of commercial alternatives**
- **2M+ Users**
- **2,500+ Applications**

Forbes
Power in the Numbers
Quentin Hardy, 05.06.10, 09:00 AM EDT
The professor who invented analytic software for the experts now wants to take it to the masses

The New York Times
Data Analysts Captivated by R's Power
By ASHLEE VANCE
Published: January 6, 2009

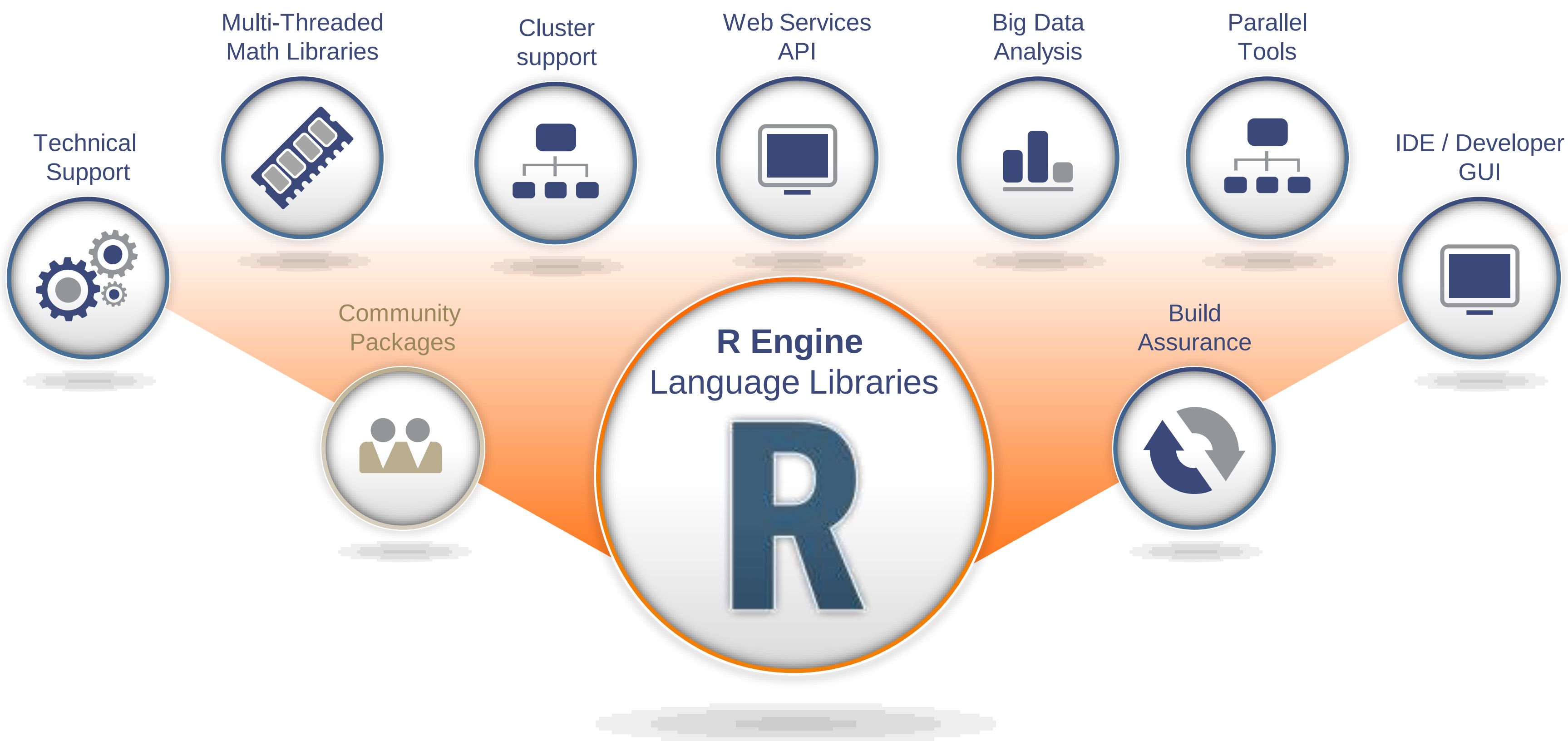
Statistics	Finance
Predictive Analytics	Life Sciences
Data Mining	Manufacturing
Visualization	Retail
	Telecom
	Social Media
	Government



Bank of America, GE, MERCK, EQUIFAX, 7-Eleven, citi, VISA, IGT, ACXIOM

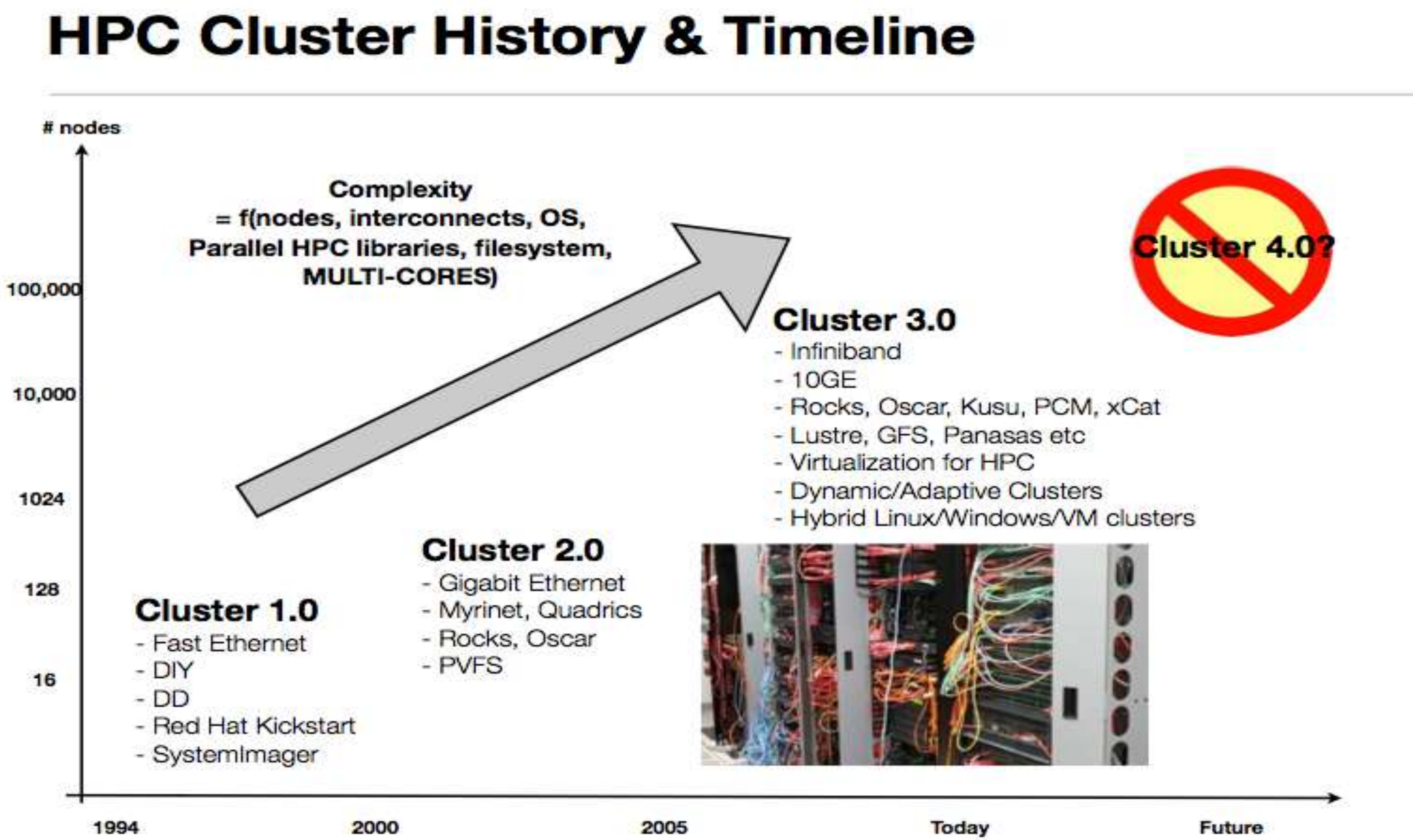
Revolution R Enterprise has Open-Source R Engine at the core

3,700 community packages and growing exponentially



WHAT IS BIG DATA AND WHY BIG DATA?

Before Cloud and Hadoop



Big Data and Analytics

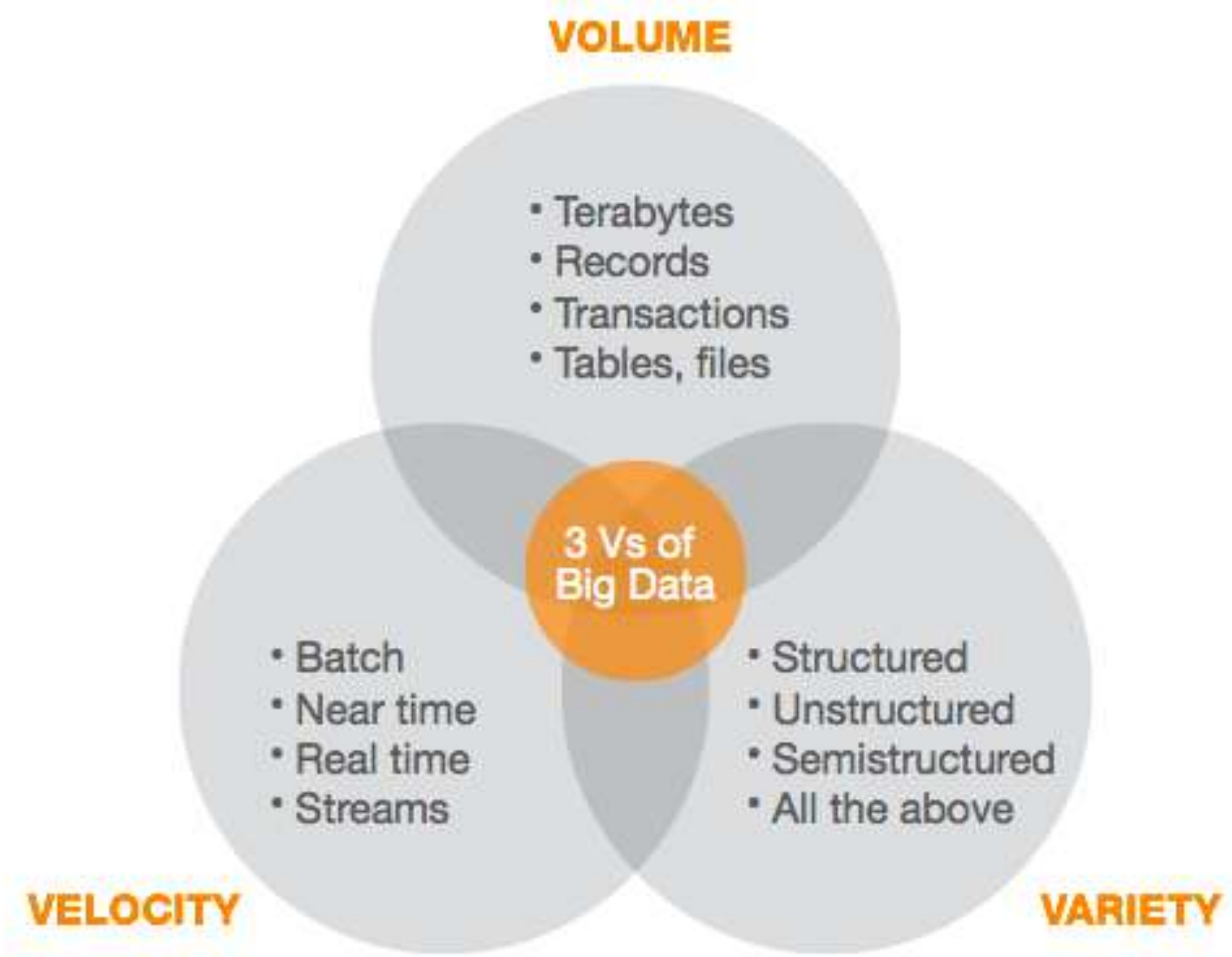
- Big Data
  - Concepts of distributing data (Hadoop) for processing is not new
    - PVFS, Lustre
    - Manual BLAST
- Analytics
  - A fancier name for Statistics???
    - Predictive analytics?
      - Neural networks? – 1980s…

Analytics = statistics + big data (social media)

What is Big Data

- "Big data" is data that becomes large enough that it cannot be processed using conventional methods..
  - BigData = f (Volume, Velocity, Variety)
- Creators of web search engines were among the first to confront this problem??
  - I beg to differ – Mapping of Human Genome in mid 2000s was the first to grapple with "big data"
- Today, social networks, mobile phones, sensors and science contribute to petabytes of data created daily.

Big Data


VOLUME
• Terabytes
• Records
• Transactions
• Tables, files
3 Vs of Big Data
• Batch
• Near time
• Real time
• Streams
• Structured
• Unstructured
• Semistructured
• All the above
VELOCITY
VARIETY

Analytics

- predictive analytics
- data mining
- statistical analysis
- complex SQL.
- data visualization
- artificial intelligence
- natural language processing
- database capabilities that support analytics
  - MapReduce
  - in-database analytics
  - in-memory databases
  - columnar data stores

Predict
Vs
Discover

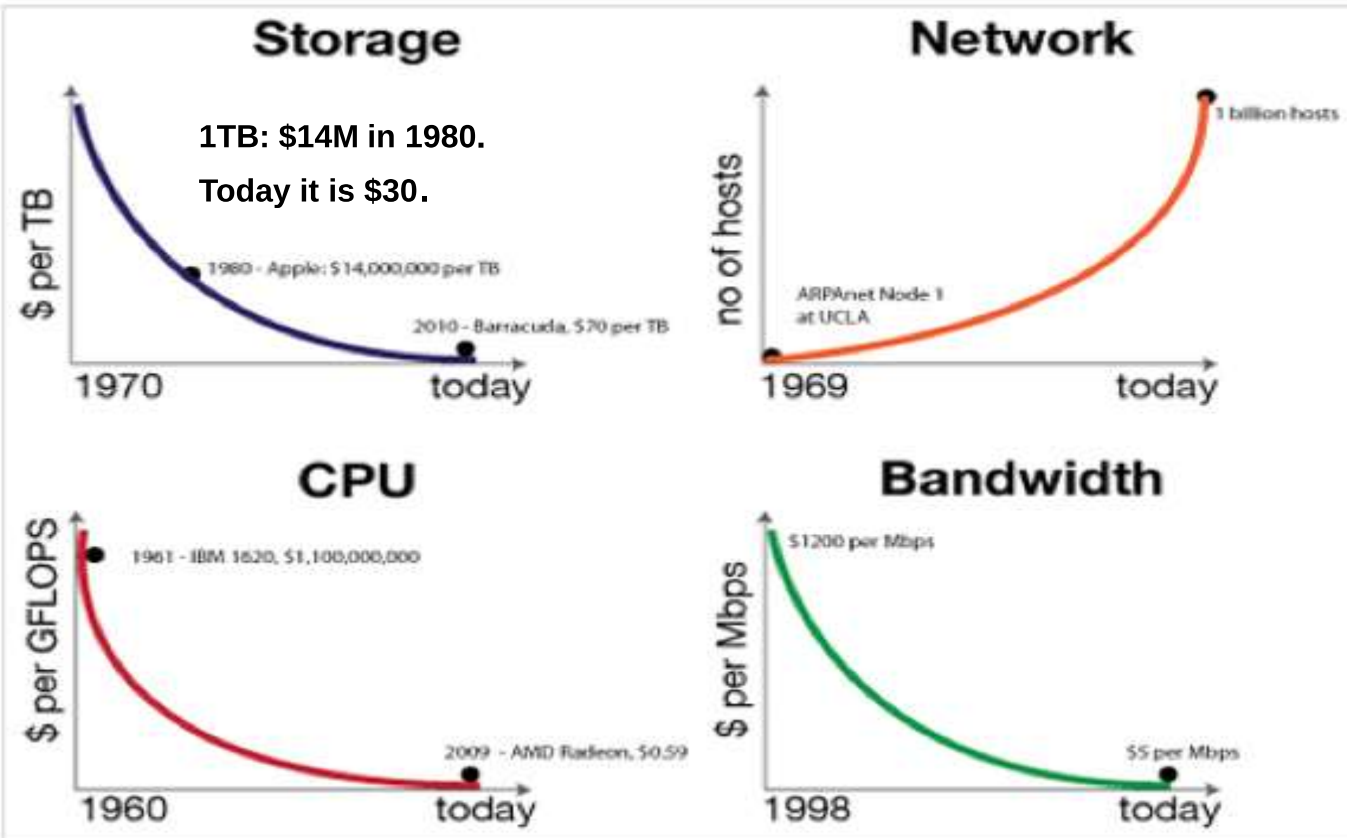
Analytics now and some best practice

NOW

Why Big Data Analytics?

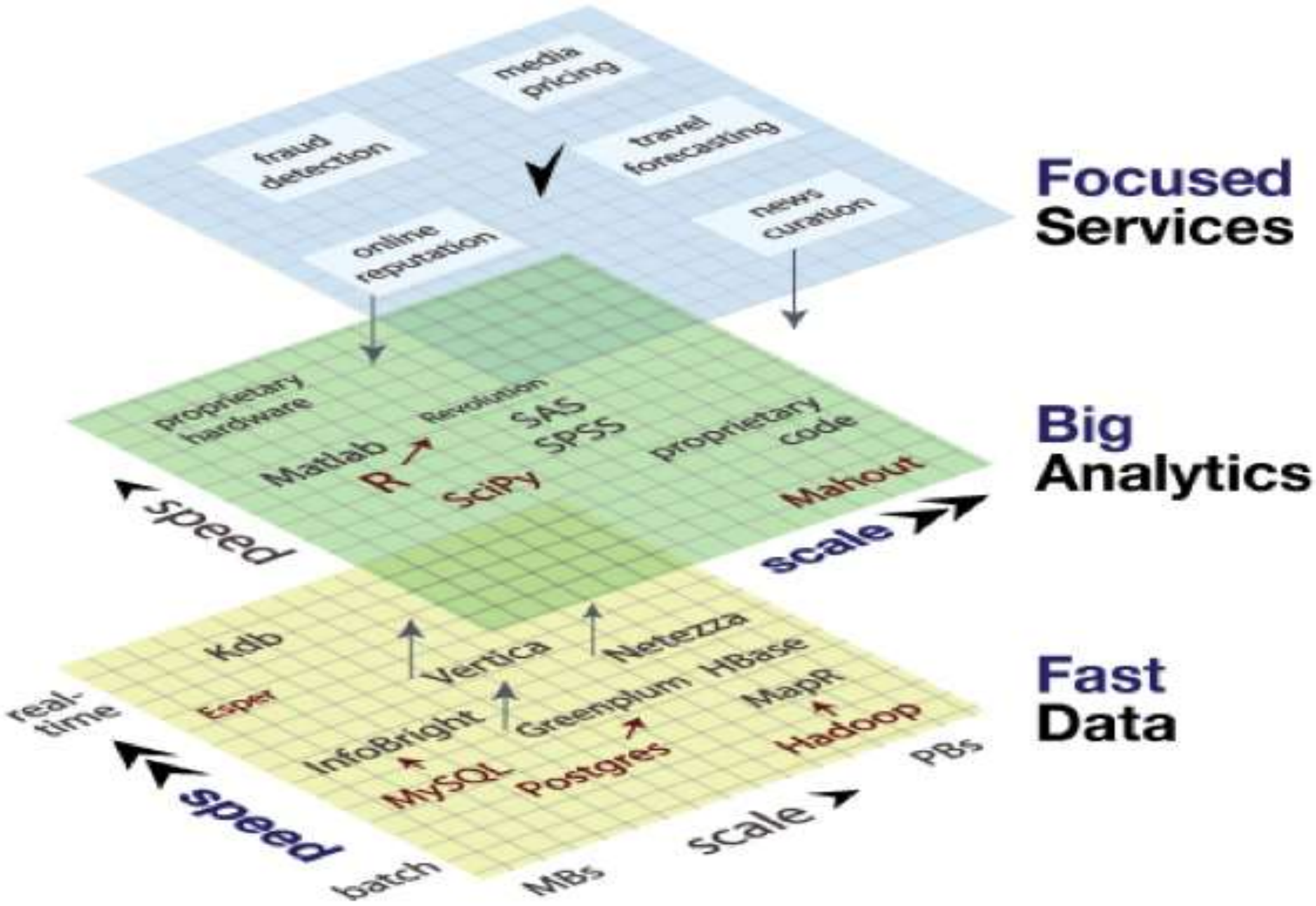
- No more sampling
- Availability of tools such as Hadoop and R
- Economics (see chart later)
- Messy data is good – as long as it's big
  - You want to know the outliers (fraud?)
  - Don't strip and clean the data
- Big Data + Analytics -> company assets with actionable business insights
  - Today it is unforgiveable to sit on data and not act on it
  - Data is treated as a perishable a good

Economics: Attack of the Exponentials



Migration to the cloud is the manifest destiny for big data, and the cloud is the launching pad for data startups.

The Emerging Big Data Stack



The R Project

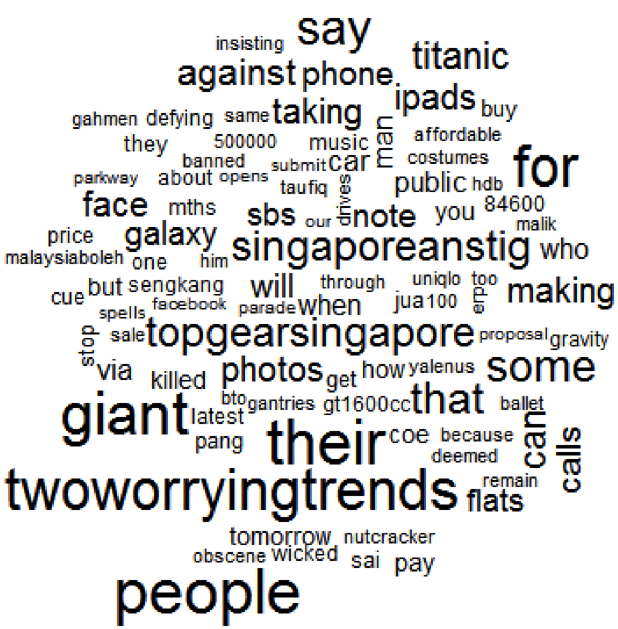
Data Analysis and Statistical Graphics
for the Enterprise

What is R?

Download the White Paper
[R is Hot](bit.ly/r-is-hot)
bit.ly/r-is-hot

- Data analysis software
- A programming language
  - Development platform designed by and for statisticians
- An environment
  - Huge library of algorithms for data access, data manipulation, analysis and graphics
- An open-source software project
  - Free, open, and active
- A community
  - Thousands of contributors, 2 million users
  - Resources and help in every domain

R Code to Create MrBrown’s WordCloud

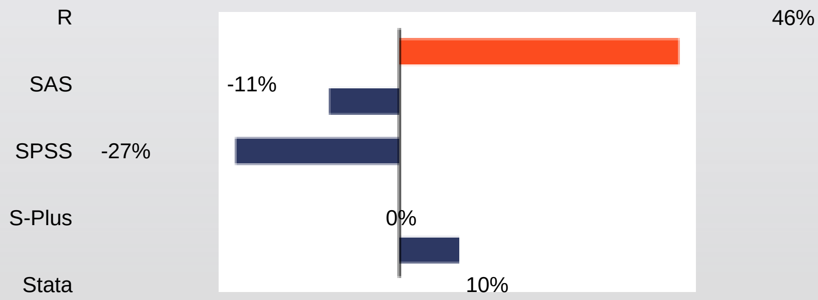



```
require(twitteR)
require(tm)

mrbrown.tweets <- searchTwitter('@mrbrown', n=1500)
text <- laply(mrbrown.tweets, function(t) t$getText())
text.corpus <- Corpus(VectorSource(text))

text.corpus <- tm_map(text.corpus, removePunctuation)
text.corpus <- tm_map(text.corpus, tolower)
text.corpus <- tm_map(text.corpus, removeWords,
c('mrbrown','english','the','with','and'))

tdm <- TermDocumentMatrix(text.corpus)
m <- as.matrix(tdm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)

wordcloud(d$word,d$freq,c(3,.3),50,150,T,.15)
```

R is exploding in popularity and functionality

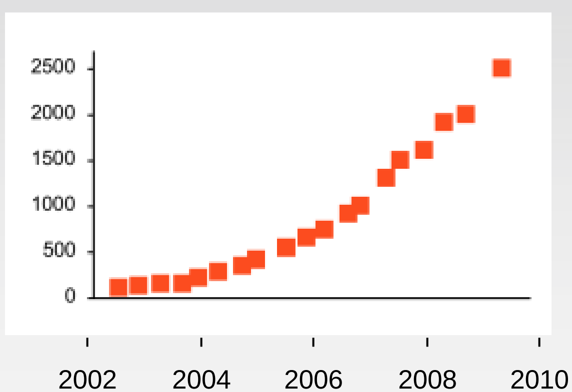
Scholarly Activity

Google Scholar hits ('05-'09 CAGR)



Package Growth

Number of R packages listed on CRAN



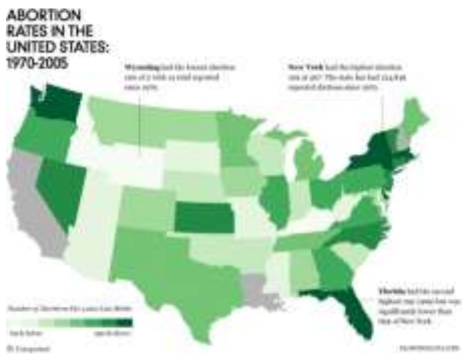
"I've been astonished by the rate at which R has been adopted. Four years ago, everyone in my economics department [at the University of Chicago] was using Stata; now, as far as I can tell, R is the standard tool, and students learn it first."

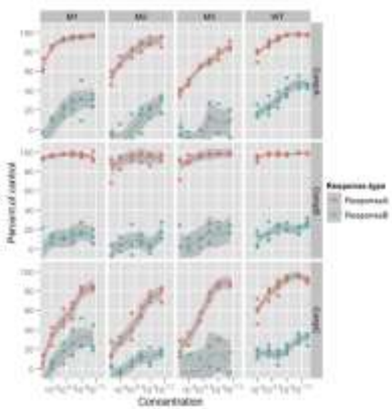
Deputy Editor for New Products at Forbes

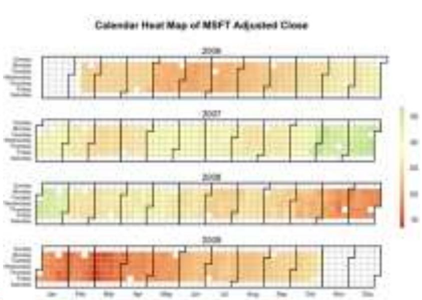
"A key benefit of R is that it provides near-instant availability of new and experimental methods created by its user base — without waiting for the development/release cycle of commercial software. SAS recognizes the value of R to our customer base…"

Product Marketing Manager SAS Institute, Inc

Graphics and Data Visualization







- Functions for standard graphs
  - Scatterplot, time series, histogram, smoothing, …
  - Bar plot, pie chart, dot chart, …
  - Image plot, 3-D surface, map, …
- Influences from Cleveland, Tufte etc.
  - Conditioning, small multiples, use of color
- Customize without limits
  - Combine graph types
  - Create entirely new graphics

Statistical Modeling

- All standard statistical methods built in
  - Mean, median, covariance, distributions, …
  - Regression, ANOVA, cross-tabulations, …
  - Survival, nonlinear mixed effects, GLM, …
  - Neural networks, trees, GAM, …
- Object-oriented functions
  - Access all parts of the analysis results
  - Combine analytic methods

Cutting-edge analytics

- Really good domain-specific suites for R:
  - Genomics: BioConductor
  - Portfolio Optimization: Rmetrics

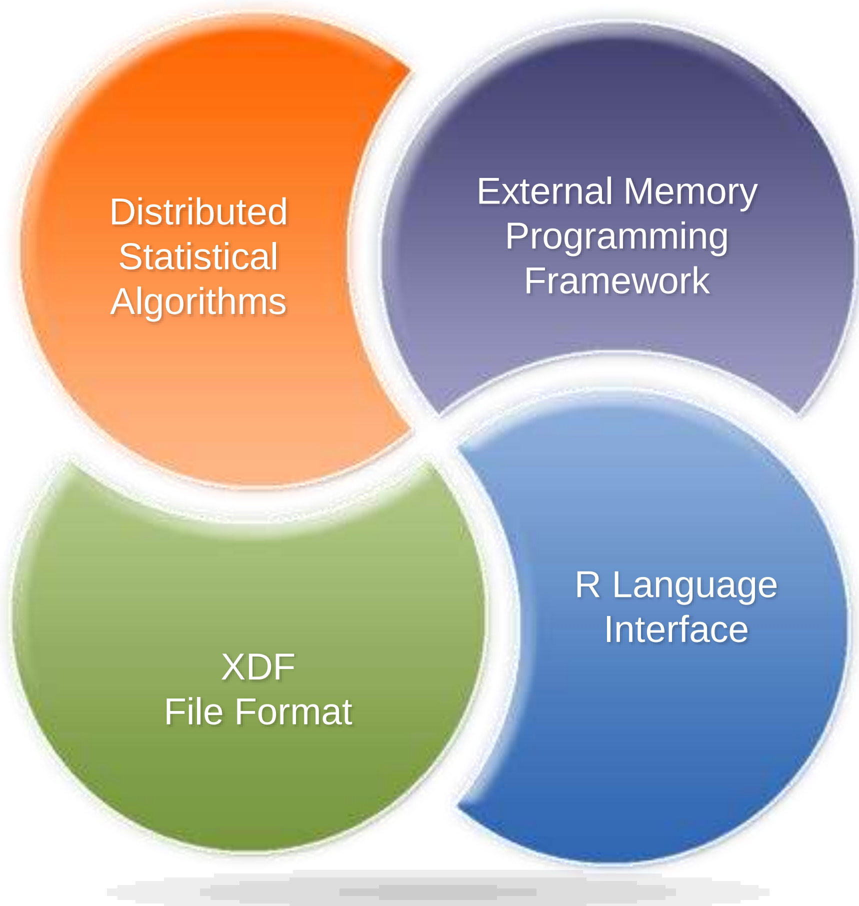
- Thousands of add-on packages:
  - CRAN: cran.r-project.org
  - Task Views
  - Machine learning, natural language processing, PK/PD, HPC, Econometrics, Environmetrics, …

HIGH PERFORMANCE ANALYTICS, BIG DATA AND HPC CLUSTERS

RevoScaleR: Big Data Analysis for Revolution R Enterprise

Addresses performance by distributing computations between cores and computers

A novel high-speed file format designed specifically to support statistical analyses



Addresses capacity through a collection of functions for chunking through massive data files

Familiar, high-prodictivity programming paradigm for R users

“I don’t have big hardware.”
Big data analysis on your desktop.

Getting Started with Big Data

- When we talk with people about their “big data”, almost always the first issue they raise is “hardware”. “What kind of hardware do I need to analyze big data.”
- My answer, “Get started today with the hardware you have.   With Revolution R Enterprise, you can quickly begin doing scalable data analysis on your desktop while you are determining your longer term hardware requirements.”

Big Data on Your Desktop

- Data sets with many variables and 100-million observations can be easily processed on a desktop using RevoScaleR functions.
- Using Revolution R Enterprise, you can **avoid getting locked into memory-bound analyses**. By processing data a chunk at a time, increasing the number of observations in your data set doesn't increase the memory requirements for a given analysis.
- There is no need to pay for $500K 1TB RAM servers!!!!!!

Estimating a Big Logistic Model

- A challenging model: a logistic regression with over 50 parameters (categorical data for Dad and Mom's ages, race, Hispanic ethnicity, live birth order, plurality, gestation, and year)

```
ItsaBoy ~ DadAgeR8 + MomAgeR7 +
    FRACEREC + FHISP_REC +
    MRACEREC + MHISP_REC +
    LBO4 + DPLURAL_REC + Gestation +
    F(DOB_YY)
```

Big Logistic Model on the Desktop

- Even a large logistic regression (over 50 parameters) with almost 100 million rows of data can be estimated on a desktop, in about the time it takes to get a cup of coffee (about 6 minutes)

Revolution R Enterprise Console

```
> system.time(
+ logitObj <- rxLogit(ItsaBoy ~ DadAgeR8 + MomAgeR7 + FRACEREC + FHISP_REC +
+      MRACEREC + MHISP_REC + LBO4 + DPLURAL_REC + Gestation + F(DOB_YY) ,
+   data=birthAll, dropFirst=TRUE, blocksPerRead = 10, reportProgress = 0 ))
   user  system elapsed
 960.53   57.46  356.77
> |
```

- But what if that's not fast enough?

“I need to be ready for tomorrow’s data.”

Scaling data analysis to a cluster.

The Birth Data Logistic Regression on a Cluster

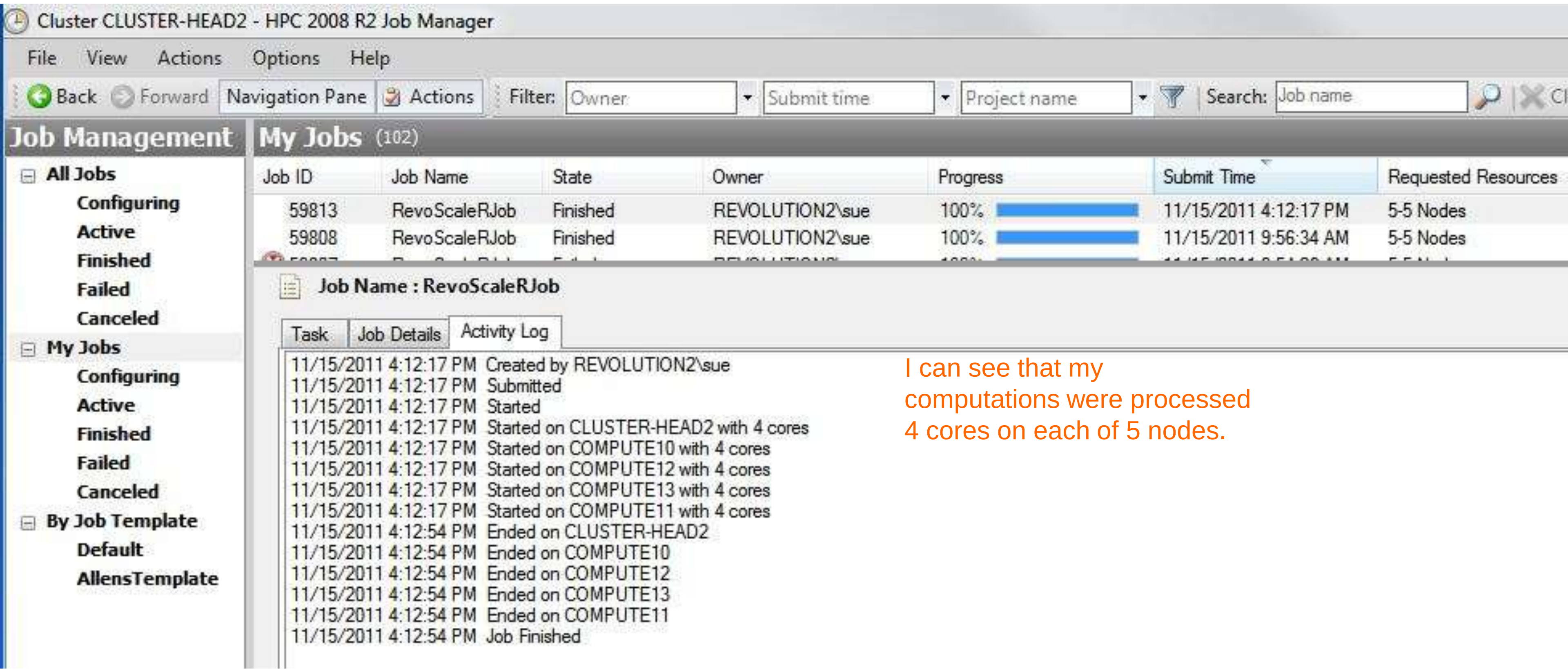
- In our office we have a 5-node cluster of commodity hardware (about $5,000) running Windows HPC Server
- I just set my compute context to use the cluster (and wait for the results) and set the location of the data on the nodes
- Then run the same code

Revolution R Enterprise Console

```
> rxOptions(computeContext = myWaitCluster)
> birthAll <- "C://data//CDC-birth//BirthUS.xdf"
> system.time(
+ logitObj <- rxLogit(ItsaBoy~ DadAgeR8 + MomAgeR7 + FRACEREC + FHISP_REC +
+     MRACEREC + MHISP_REC + LBO4 + DPLURAL_REC + Gestation + F(DOB_YY) ,
+   data=birthAll, dropFirst=TRUE, blocksPerRead = 10, reportProgress = 0 ))
   user  system elapsed
   0.59    0.00   41.61
>
```

42 seconds instead of 6 minutes

HPA Jobs on a Windows HPC cluster



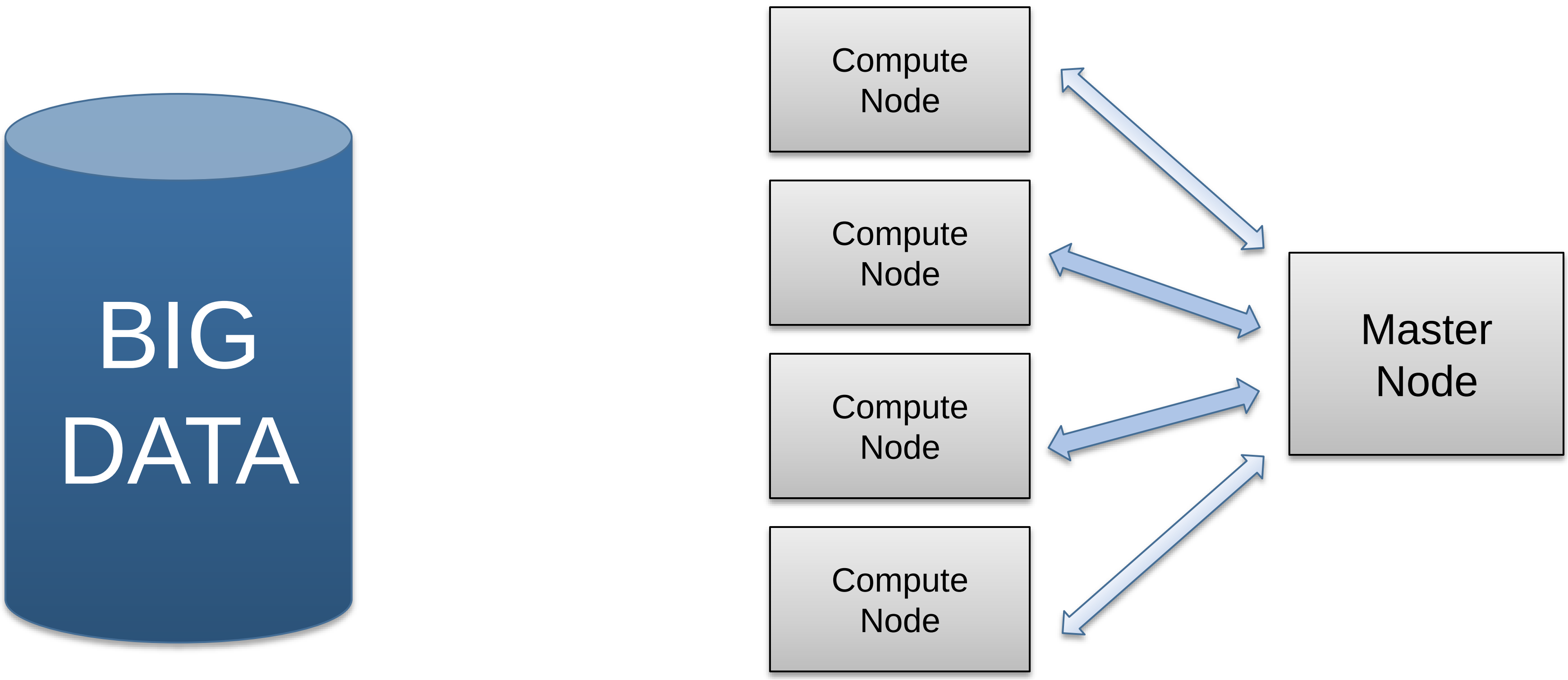
I can see that my computations were processed 4 cores on each of 5 nodes.

HPA Benchmarking comparison* – Logistic Regression

	Competitor	Revolution Analytics
Rows of data	1 billion	1 billion
Parameters	"just a few"	7
Time	80 seconds	44 seconds
Data location	In memory	On disk
Nodes	32	5
Cores	384	20
RAM	1,536 GB	80 GB

Revolution R is faster on the same amount of data, despite using approximately a 20th as many cores, a 20th as much RAM, a 6th as many nodes, and not pre-loading data into RAM.

RevoScaleR Big Data Analytics
Servers & Distributed Clusters



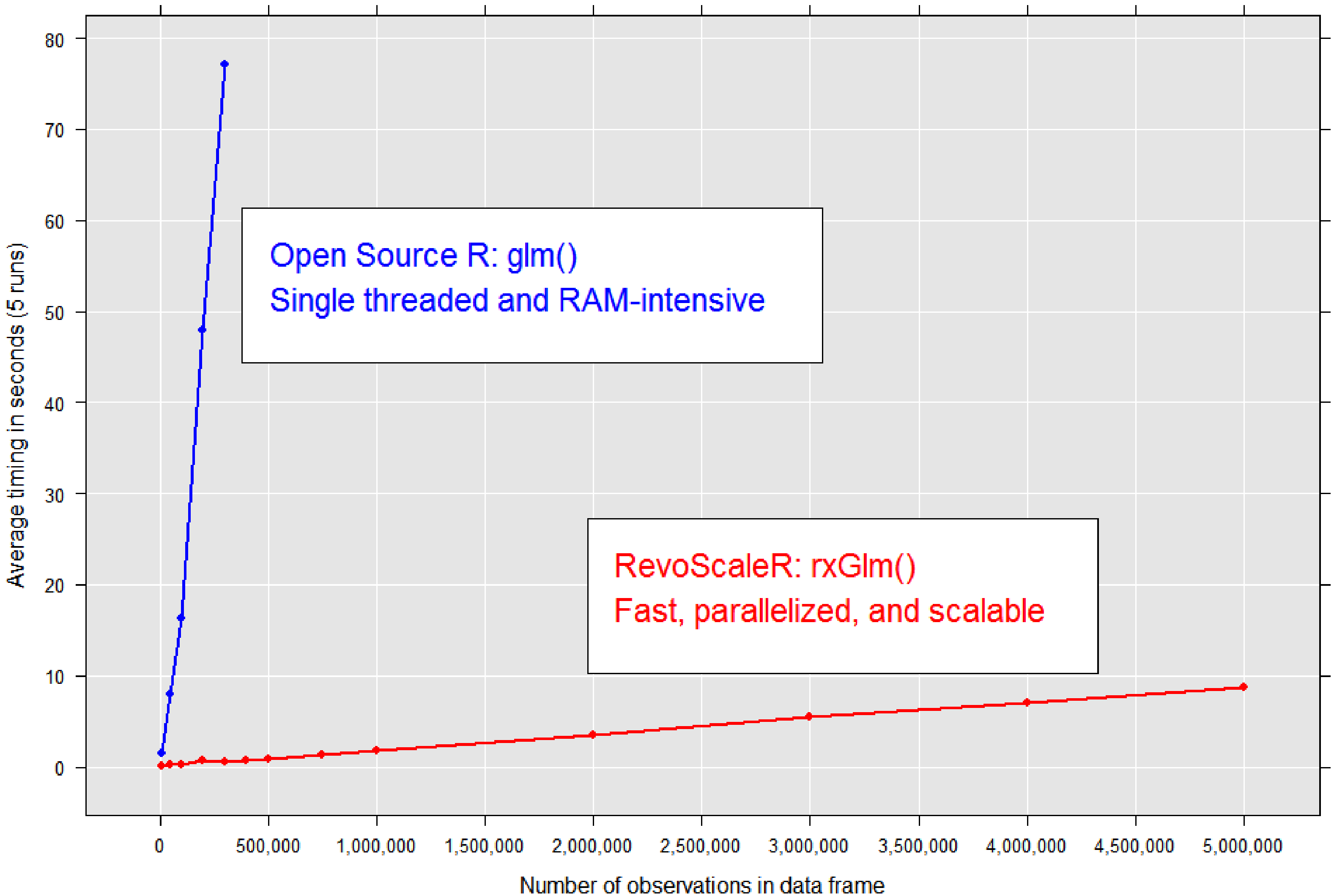
- Data Step, Statistical Summary, Tables/Cubes, Covariance, Linear & Logistic Regression, GLM, K-means clustering, …

Common Analytic Platform across Big Data Architectures

R

Hadoop	File Based Cluster	In-database
✓ IBM InfoSphere BigInsights	✓ Platform Computing IBM	✓ Netezza
✓ cloudera	✓ Windows HPC Server 2008	Teradata
Hortonworks		Greenplum
MapR Technologies		

GLM 'Gamma' Simulation Timings
Independent Variables: 2 factors (100 and 20 levels) and one continuous



Timings from a Windows 7, 64-bit quadcore laptop with 8 GB RAM

Paradigms for Statistical Analysis – High Performance Computing (HPC)

(embarrassingly parallel)

- The purpose of HPC-type analytics is to generate many "answers" that are independent from one another.
  - Parallel independent execution of an R function across cores and nodes
  - Usually involve small amounts of data (such as an individual's credit history within a very large aggregate amount of data for an entire population)
  - Some Examples:
    - Scoring
    - Simulations (Monte Carlo)
    - Binning of data for visualizations

Paradigms for Statistical Analysis – High Performance Analytics (HPA)

(tightly coupled)

- The purpose of HPA-type analytics is to generate a single "answer"
  - There is more data than fits into memory and the model requires that you use all the data to get the answer
  - The calculation is broken into small interim steps whose results are assembled into a final result
  - Algorithms are parallelized to execute across cores and nodes (Parallelized External Memory Algorithms)
  - Executions are dependent of each other
  - Some Examples:
    - Linear regression
    - Logistic regression
    - Kmeans Clustering

RevoScaleR: Big-Data Algorithms

Big-Data Algorithm	Example Applications	Revolution Analytics
Data Step	ETL, data distillation, record/variable selection, variable transformation	✓✓
Descriptive Statistics	Exploratory Data Analysis, Data Validation	✓✓
Tables & Cubes	Reporting, contingency analysis	✓✓
Correlation / Covariance	Factor Analysis, Value at Risk	✓✓
Linear regression	Forecasting, Net present value estimation	✓✓
Logistic Regression	Response modeling, offer selection	✓✓
Generalized Linear Models	Capital reserve estimation, climate modeling	✓✓
K-means clustering	Customer Segmentation	✓✓
Model Prediction	Real-time Scoring (decisions, offers, actions)	✓✓
Parallel & distributed computing with R	Simulations, By-Group analysis, ensemble models, custom applications	✓✓

Revolution Analytics Distributed Computing Implementations

- For HPC-Type on:
  - Linux/MS HPC Clusters – RevoScaleR, using **rxExec**
  - IBM Netezza, using **nzApply** and **nzTApply** (**nzr** package)
  - Hadoop – **mapreduce** in **rmr** package using only a map function
- For HPA-Type on:
  - Linux/MS HPC Clusters – RevoScaleR, using **rxLinMod, rxLogit, rxCube**…
  - IBM Netezza, using **nzLm**, **nzKMeans**… (**nza** package)
  - Hadoop – **mapreduce** in **rmr** package. Requires custom R scripting.

R AND HADOOP

Hadoop

Apache Hadoop is an open source platform for data storage and processing that is…

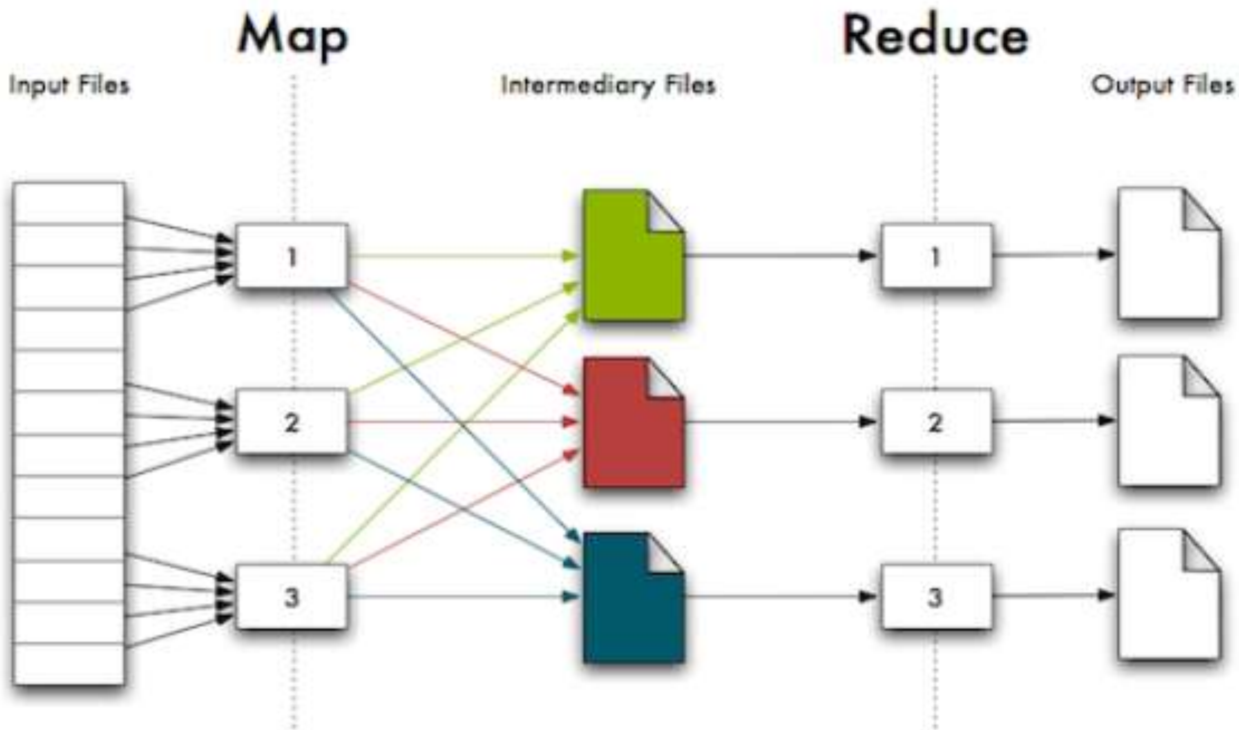
- ✓ Scalable
- ✓ Fault tolerant
- ✓ Distributed

CORE HADOOP SYSTEM COMPONENTS

Hadoop Distributed File System (HDFS)	+	MapReduce
Self-Healing, High Bandwidth Clustered Storage		Distributed Computing Framework

Provides storage and computation in a single, scalable system.

Mapreduce



- MapReduce distributes jobs across nodes of the Hadoop cluster
- The Map function operates on a block of data and produces intermediate output
- The Reduce function takes the intermediate output and aggregates it into a final set of results.
- MapReduce jobs can be written in R, Pig, Java, Python and other languages

R and Hadoop

- **Hadoop** offers a scalable infrastructure for processing massive amounts of data
  - Storage – HDFS, HBASE
  - Distributed Computing - MapReduce
- **R** is a statistical programming language for developing advanced analytic applications
- **Currently**, writing analytics for Hadoop requires a combination of Java, pig, Python, …
- **The Rhadoop project** makes it possible to write Big Data algorithms for Hadoop using the R language alone.

Motivations for Rhadoop Project

- **Make** it easy for the R programmer to interact with the Hadoop data stores and write MapReduce programs
- **Ability** to run R on a massively distributed system without having to understand the underlying infrastructure
- **Keep** statisticians focused on the analysis and not the implementation details
- **Open** source to drive innovation and collaboration.

A Growing Market with Affinity for R

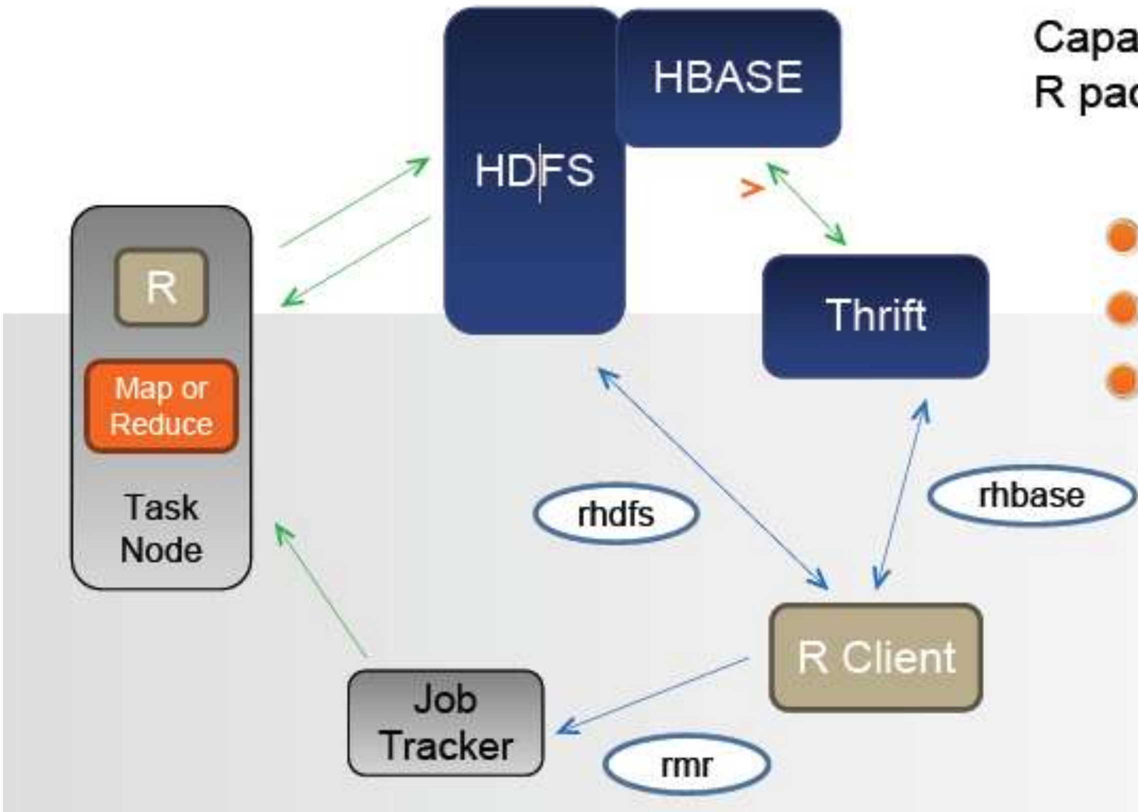
- IDC estimates Hadoop software market to reach $812M by 2016
- Lots of experimentation being led by IT
- Hadoop is particularly well-suited for unstructured data; the fastest-growing type
- R was an early player in the Hadoop ecosystem
- Hadoop is a catalyst for analytics platform re-engineering
  - driving R use even in established SAS shops

REVOLUTION ANALYTICS HADOOP CAPABILITIES

Rhadoop Project

- Revolution Analytics conceived, engineered and built the packages in the RHadoop Project
- New releases available approximately every quarter
- RHadoop Project consists of 3 R packages
  - rhdfs – connector from R to HDFS (read/write)
  - rhbase– connector from R to HBASE (read/write)
  - rmr– execute MapReduce jobs written 100% in R

R and Hadoop – The R Packages

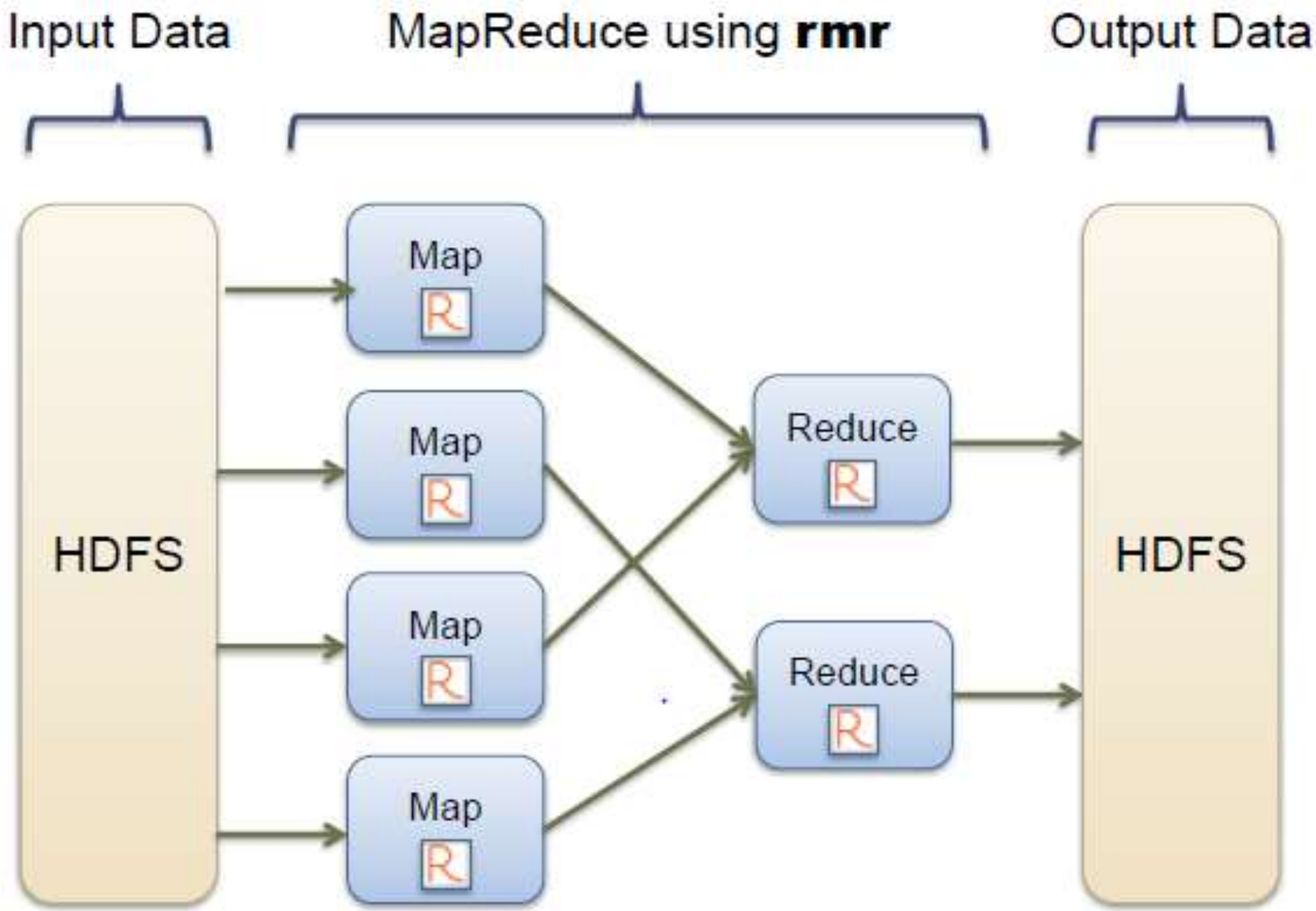


Capabilities delivered as individual R packages

- rhdfs - R and HDFS
- rhbase - R and HBASE
- rmr - R and MapReduce

Downloads available from Github

RHadoop – MapReduce Using rmr



When Working with Hadoop, Both Steps of Data Analysis Can Use MapReduce with rmr

- Data Distillation/ Data Step
  - rmr can be used within Hadoop to extract meaning from unstructured data
    - Create new variables such as counts (e.g. number of clicks in a day)
    - sort (e.g. according to criteria or sentiment)
    - merge
  - These sorts, merges, new variables, etc. can either be used within Hadoop for analytics or can be pulled into Revolution R Enterprise for statistical analysis

- Statistical Analysis within Hadoop
  - HPC-type analytics can be executed using rmr and R functions
  - HPA-type analytics can be executed using rmr via custom R scripting.
    - A library of RevoScaleR HPA routines for Hadoop is coming

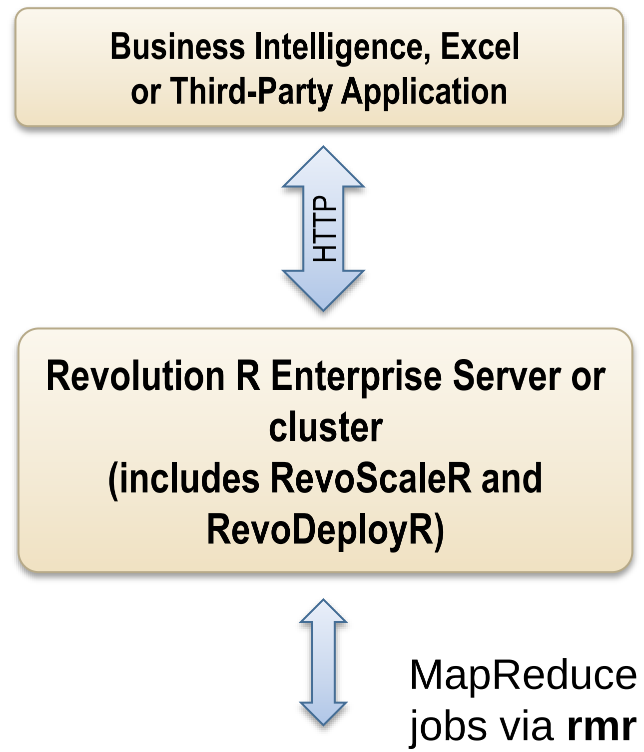
Two Basic Deployment Models

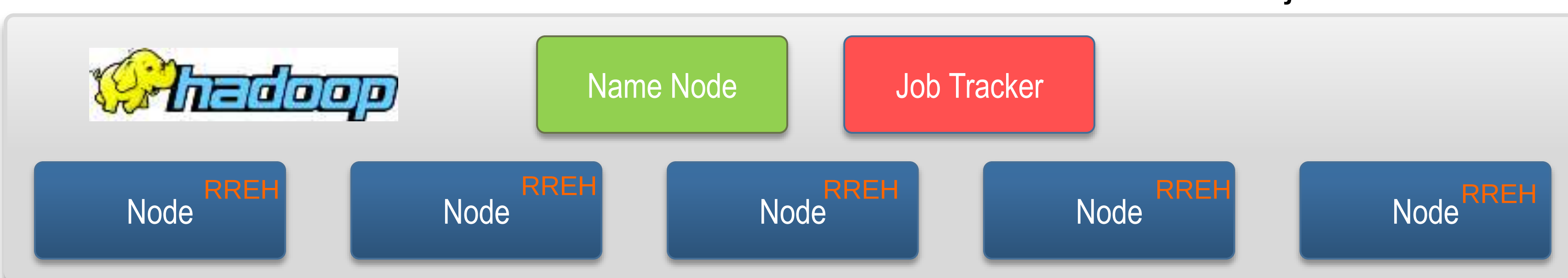
- Option 1 – rmr in use. **Revolution R Enterprise next-to** and **Revolution R for for Hadoop** installed ***Inside*** Hadoop to provide both:
  - rmr-enabled statistical analytics within Hadoop
  - rmr-enabled data distillation within Hadoop for statistical analyses inside or next to Hadoop

- Option 2 – rmr not in use. **Revolution R Enterprise installed next-to** Hadoop to provide:
  - rhdfs- and rhbase-based access to Hadoop as a data source for HPA
  - statistical analysis done in Revolution R Enterprise on one or more edge nodes.

Option 1 : rmr in use

- Data distillation or Statistical Analysis are run as a MapReduce job in the Hadoop Cluster via **rmr**
- Standalone Revolution R Enterprise “client” server (or cluster) is physically connected to Hadoop cluster and is used to:
  - Kick off MapReduce jobs using rmr
  - Access Hadoop jar files (i.e. can submit a job to the job tracker)
  - Connect to Hadoop data stores (HDFS and HBASE) using rhdfs or rhbase packages
  - Build and test models
  - Collect results for further processing, visualization
  - Propagate results through RevoDeployR.
  - RREH Is Revolution R Enterprise for Hadoop

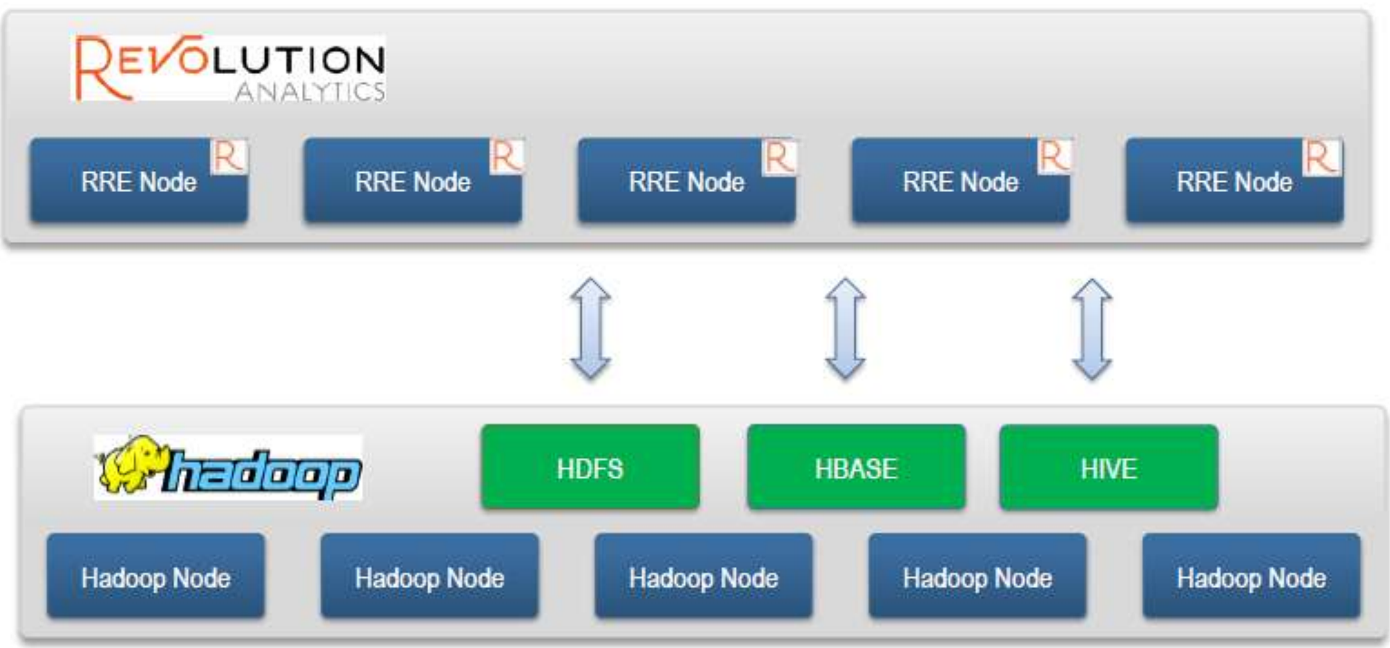




REVOLUTION ANALYTICS

Option 2 – no rmr in use

- Hadoop data accessed from RRE using rhbase, rhdfs, RODBC. We assume that rmr has not been used to distill / prepare the data
- Statistical analytics processing is on separate server or shared cluster using Revolution R Enterprise



rhdfs

- Manipulate HDFS directly from R
- Mimic as much of the HDFS Java API as possible
- Examples:
  - Read a HDFS text file into a data frame.
  - Serialize/Deserialize a model to HDFS
  - Write an HDFS file to local storage
  - rhdfs/pkg/inst/unitTests rhdfs/pkg/inst/examples

rhdfs Functions

- File Manipulations
  - hdfs.copy, hdfs.move, hdfs.rename, hdfs.delete, hdfs.rm, hdfs.del, hdfs.chown, hdfs.put, hdfs.get
- File Read/Write
  - hdfs.file, hdfs.write, hdfs.close, hdfs.flush, hdfs.read, hdfs.seek, hdfs.tell, hdfs.line.reader, hdfs.read.text.file
- Directory
  - hdfs.dircreate, hdfs.mkdir
- Utility
  - hdfs.ls, hdfs.list.files, hdfs.file.info, hdfs.exists
- Initialization
  - hdfs.init, hdfs.defaults

rhbase

- Manipulate HBASE tables and their content
- Uses Thrift C++ API as the mechanism to communicate to HBASE
- Examples
  - Create a data frame from a collection of rows and columns in an HBASE table
  - Update an HBASE table with values from a data frame
  - rhbase/pkg/inst/unitTests

Rhbase Functions

- Table Manipulation
  - hb.new.table, hb.delete.table, hb.describe.table, hb.set.table.mode, hb.regions.table
- Row Read/Write
  - hb.insert, hb.get, hb.delete, hb.insert.data.frame, hb.get.data.frame, hb.scan
- Utility
  - hb.list.tables
- Initialization
  - hb.defaults, hb.init

rmr

- Designed to be the simplest and most elegant way to write MapReduce programs
- Gives the R programmer the tools necessary to perform data analysis in a way that is “R” like
- Provides an abstraction layer to hide the implementation details
- Examples
  - Simulations - Monte Carlo and other Stochastic analysis
  - R ‘apply’ family of operations (tapply, lapply…)
  - Binning, quantiles, summaries, crosstabs and inputs to visualization (ggplot, lattice).
  - Machine Learning
  - rmr/pkg/inst/tests

rmr mapreduce Function

- mapreduce (input, output, map, reduce, …)
  - input – input folder
  - output – output folder
  - map – R function used as map
  - reduce – R function used as reduce
  - … - other advanced parameters

RHADOOP – THE BASICS

Simple Example

```
small.ints <- 1:10
out <- lapply(small.ints, function(x) x^2)

small.ints <- to.dfs(1:10)
out <- mapreduce(input = small.ints,
                map = function(k,v) keyval(v, v^2))
```

Binomial Example

```
Groups <- rbinom(32, n = 50, prob = 0.4)
out <- tapply(groups, groups, length)

groups <- to.dfs(groups)
out <- mapreduce(input = groups,
          map = function(k, v) keyval(v, 1),
          reduce = function(k,vv) keyval(k, length(vv)))
```

Wordcount

```
wordcount <- function(input, output = NULL, pattern = " ")
{
   mapreduce(input = input , output = output,
    input.format = "text",
    map = function(k,v)
    {
        lapply( strsplit( x = v,
            split = pattern)[[1]],
            function(w) keyval(w,1))
    },
    reduce = function(k,vv)
    {
        keyval(k, sum(unlist(vv)))
    }, combine = T)
}
```

Logistic Regression

```
logistic.regression <- function(input, iterations, dims, alpha)
{
    plane <- rep(0, dims)
    g <- function(z) 1/(1 + exp(-z))
    for (i in 1:iterations)
    {
        gradient <- from.dfs(mapreduce(input,
            map = function(k, v) keyval (1, v$y * v$x * g(-
                v$y * (plane %*% v$x))),
            reduce = function(k, vv) keyval(k,
                apply(do.call(rbind,vv),2,sum)),
            combine = T)) plane = plane + alpha *
                gradient[[1]]$val
    }
    plane
}
```

K-means

```
kmeans <-
    function(points, ncenters, iterations = 10,
        distfun = function(a,b) norm(as.matrix(a-b), type='F'))
        {
            newCenters <- kmeans.iter(points, distfun = distfun,
                    ncenters = ncenters)
            for(i in 1:iterations)
            {
                newCenters <- lapply(values(newCenters), unlist)
                newCenters <- kmeans.iter(points, distfun, centers
                    = newCenters)
            }
        newCenters
        }
```

K-means

```
kmeans.iter <-
    function(points, distfun, ncenters = length(centers), centers = NULL)
    {
        from.dfs(
            mapreduce(input = points, map = if (is.null(centers))  {
                function(k, v) keyval(sample(1:ncenters,1), v)
            } else {
                function(k, v) {
                distances <- lapply(centers, function(c) distfun(c, v))
                keyval(centers[[which.min(distances)]], v)
                }
            },
    reduce = function(k, vv) keyval(NULL, apply(do.call(rbind, vv),
            2,mean))))
}
```

Is Hadoop 2.0/ARN the right platform for you?

- In terms of YARN, the OMPI-based "HOD" solution launches an MPI program about 1000x faster, and runs about 10x faster. The launch time differences grows with scale as the YARN MPI solution wires up with a quadratic time signature, while the OMPI solution wires up logarithmically.
- The execution time difference depends upon the application (IO bound vs compute bound), but largely stems from a difference in available data transports.
- As a practical example, running a simple MPI "ring" program takes about 90 seconds on an 8 node system using YARN, and about 35 milliseconds using OMPI under SLURM.
- An MR word count program that looked at 1000 files took about 6 minutes using YARN, and about 11 seconds using OMPI's MR+.
- Non-MPI programs also tend to launch faster due to the difference in how YARN handles launch vs other RMs.
- Again, a non-MPI "hello" running on an 8 node system can still take 20 seconds to run, depending on the heartbeat setting, and about 25 milliseconds using SLURM.

Future: Diverging data paradigms

More data, better fault tolerance →

Files & Clusters

scale R

Platform Computing

Windows HPC Server 2008

Databases

NETEZZA

EMC²

TERADATA Raising Intelligence

GREENPLUM

Hadoop

cloudera

hadoop

Exploration
Modeling

← Easier programming, better performance

Storage
Preprocessing

Production

Final thoughts on RHadoop

- R and Hadoop together offer innovation and flexibility needed to meet analytics challenges of big data
- Connects the R Programmer and the Hadoop Expert
- We need contributors to this project!
  - Developers
  - Documentation
  - Use cases
  - General Feedback