Divide & Recombine (D&R), RHIPE, and Riposte for Large Complex Data

The Mozilla Corporation
Saptarshi Guha

Statistics, Purdue
Ashrith Barthur
Bill Cleveland
Philip Gautier
Xiang Han
Jeff Li
Jeremy Troisi
Bowei Xi

PNNL
Ryan Hafen

Computer Science, Stanford
Zack DeVito
John Gerth
Pat Hanrahan
Justin Talbot

Computer Science, U. of Washington
Jeff Heer

Computer Science Purdue
Jan Vitek

GE Global Research
Jin Xia

Large Complex Datasets

Ubiquitous today

Currently challenge everything involved in the analysis of data

- numeric methods of statistics and machine learning
- numeric models of statistics and machine learning
- visualization methods
- computational algorithms
- computational environments: hardware and software

Defining the Community and the Analysis

Scientists, engineers, business analysts, etc.

Use of methods of statistics, machine learning, and visualization

Collectively, need to bring to bear the 1000s of methods of stat, ml, and vis

How can we develop an approach to analysis and a computational environment that serves them all

Two Goals

1. Deep Analysis

Definition

- comprehensive detailed analysis
- does not lose important information in the data

To achieve must use

- numeric methods of statistics and machine learning
- visualization methods
- need to visualize the detailed raw data through the whole analysis, starting with raw data

Two Goals

2. Interactive Language for Data Analysis (ILDA)

The data analyst and the methodologist developing methods for large data work within an ILDA

An example of a very effective ILDA is R

A very effectively designed ILDA provides very time-efficient programming for

- data analysis
- prototyping new methods and models

An ILDA can provide access to the 1000s of numeric and visualization methods, necessary for deep analysis

Size vs. Complexity

Sheer size, per se, is not the only challenge of large complex data to achieving deep analysis and using an ILDA

Size often not directly the challenge

With growing size typically comes a growing

- complexity of data structures
- patterns in the data
- the models needed to account for the patterns

It is the complexity that typically challenges methods, models, and computational environments

Data sets as small as 100s of gigabytes can challenge deep analysis

Achieving the Two Goals: Divide and Recombine (D&R)

D&R is an approach to the analysis of large complex data to meet the challenges

The data are parallelized: divided into subsets in one or more ways

Numeric and visualization methods are applied to each of the subsets separately

Then the results of each method are recombined across subsets

Computation is “embarrassingly parallel” and can be carried out on a cluster with nodes that can vary substantially in performance

Can exploit distributed computational environments like Hadoop = HDFS + MapReduce

D&R Enables Comprehensive Visualization for Large Complex Data

Visualization of the detailed raw data as well as summaries is critical for for deep analysis

Made feasible by using statistical sampling and experimental design

Choose a representative or focused sample of the subsets: apply visualization methods to each subset in the sample

The visualization is a visual recombination

Sampling guided by between-subset variables (each has one value per subset) that are computed

D&R Enables Use of Almost Any Numeric Method of Analysis

Final result for an numeric analytic method

Generally not the same as the result had it been feasible to apply the method directly to all of the data

D&R Enables Use of Almost Any Numeric Method of Analysis

Statistical theory and methods for D&R

The division and recombination methods have an immense impact on the accuracy of the D&R result

Smart choices of division and recombination methods can result in excellent statistical accuracy, sometimes very close to that of all-data direct computation

We are developing division methods and recombination methods that are “optimal” given data must be divided

Reduce statistical efficiency as little as possible

D&R Research

D&R research consists of the development of

- statistical methods for division
- statistical methods for recombination
- a new statistical theory whose concept is optimality subject to the constraint of having to break up the data
- computational algorithms
- computational environments

Computational Environments for D&R

Riposte and RHIPE

- novel software systems
- developed to meet the computational challenges
- make D&R computationally feasible

Hadoop Distributed Computational Environment

Distributed file system (HDFS)

Distributed compute engine (MapReduce)

Supported by Apache Software Foundation

Open source (Apache License)

History

- Google: released the architecture in a paper
- Yahoo: first public domain release of software
- Apache Software Foundation: signed on to develop and distribute

RHIPE: R and Hadoop Integrated Programming Environment

[hree-**pay**]: “in a moment” in Greek

First developed by Saptarshi Guha, formerly Purdue, now at Mozilla

Second member of core development team, Jeremiah Rounds, Purdue

A merger of R and Hadoop

Written in C, Java, and R

RHIPE: How It Works for Data Analysis or Methodology/Model Development

Program entirely within R

Saves time-costly programming in a lower-level language

RHIPE: How It Works for Data Analysis or Methodology/Model Development

Division: Creating Subsets in the HDFS

The data analyst or methodologist specifies subsets of the data using R commands

R code passed to RHIPE R commands, which create the subsets and tell Hadoop to write them to HDFS

In our work the number of subsets has ranged from 540 to 14,161,628

HDFS spreads subsets across the cluster in 128MB blocks transparent to user

These computations are elephants

RHIPE: How It Works for Data Analysis or Methodology/Model Development

Parallel Computing of R Code across Subsets

The data analyst or methodologist writes R code for a method to be applied to each subset

R code passed to RHIPE R commands that tell Hadoop to execute the code on each subset and write the results to HDFS

Hadoop schedules subset computations optimally across cluster and writes results to HDFS, transparent to user

Results can stay in HDFS for further distributed computation

These computations are elephants

RECOMBINATION

Results from distributed computation that are data reductions often can be read into the standard R file system

Analyze the results using the traditional R computing

- the analyst or methodologist operates in a highly interactive mode
- issues sequences of commands
- many compute virtually instantaneously: output is needed virtually instantaneously

Recombination is almost always carried out by such computations

These computations are mice

RHIPE Performance

Running designed experiments to study the complex system: R, RHIPE, Hadoop, and hardware

Results of one of our clusters, used only for D&R research

- 11 Hewlett Packard ProLiant DL165 G7 servers each with
- dual 2.1 GHz 12-core AMD Opteron 6172 processors (24 cores), and 22 cores reserved for Hadoop
- 48 gigabytes of memory
- 8 terabytes of local disk
- 10 gigabit Ethernet interconnect

RHIPE Performance

R `glm.fit` for logistic regression

- $n = 2^{30}$ observations, about 1 billion
- $2^7 - 1 = 127$ explanatory variables
- 2^{30} numeric values of of 2^7 variables
- $2^3 = 8$ bytes per numeric
- data size = 2^{40} bytes, about 1 terabyte
- Number of subsets = 2^{20}, about 1 million

Elapsed time in R using RHIPE/Hadoop = 1056 sec = 17.6 min

70% of this time was simply the once only R read of the subsets from the HDFS into memory!

Isn’t This Just MapReduce?

D&R exploits the MapReduce computational system implemented in Hadoop

Nothing in this system brings a focus to how a data analyst can best

- 1. divide the data (“chunk” in MapReduce parlance)
- 2. recombine the output (“reduce” in MapReduce parlance)

D&R adds a statistical layer to figure this out

D&R seeks smart chunking and reducing

Technology Transfer: RHIPE

Open source software, Apache license

R cran package: RHIPE

Google discussion group: groups.google.com/group/rhipe

Code development site: code.google.com/p/rhipe

See datadr.org

Riposte: A Fast R Virtual Machine

[ri-**pohst**] “a quick, sharp return in speech or action; counter-stroke”

R, like Javascript, is a very dynamic language

Many of the recent developments for Javascript can enable dramatic improvement in R performance

Testing has suggested 5-10 times performance improvement is possible

Riposte

New “from scratch” R virtual machine designed for performance

Key Design Insight

R contains two very different language components

- a high productivity scripting language (like JavaScript)
- a high performance vector language (like APL)

Riposte VM has 2 cooperative subVMs

- handle the scripting and vector portions of the language separately
- permits aggressive optimization of each

Riposte Performance

Early performance results

”Scripting-like” code

- 2-4x faster in Riposte than in R’s recent bytecode interpreter
- lots of low-hanging fruit left

Vector heavy code is 5-10x faster on a single core

FODAVA Work on D&R Led to Two New Grants

DARPA: XDATA.

- Title: D&R for Large Complex Data: Statistics, Visualization, and Computational Environments
- 10/01/2012 – 03/31/2014, renewable through 2017
- Pat Hanrahan (CS, Stanford). Bill Cleveland (Statistics, Purdue). Ryan Hafen (Statistics, PNNL). Jeff Heer (CS, U. of Washington)

NSF: DMS CDS&E

- Statistical Theory and Methods for D&R Analysis of Large Complex Data
- 09/01/2012 – 08/31/2015.
- Bill Cleveland, Chuanhai Liu, and Bowei Xi (Statistics Purdue)