“BIG DATA”
WHAT DOES IT MEAN FOR ME?

Amar Nath Gupta        Bill West
San Diego Supercomputer Center
University of California San Diego
ASSUMPTIONS ABOUT THE AUDIENCE

• Everyone does some sort of science
• Everyone knows some programming at some level but are not experts
• Most people perform some kind of activity with data
  • Gathering
  • Storage
  • Organizing
  • Querying and searching
  • Computing and analysis including statistics, data mining and predictive modeling
• Everybody uses some, possibly discipline or task-specific, software to accomplish their data activity
A BIT OF BACKGROUND

• We work with a number of “data scientists” from various disciplines
• Who is a data scientist?
  • Data scientists solve complex data problems through employing deep expertise in some scientific discipline
  • The general goal is to operate on data, perform some analysis and in general, gain some insight from the data
  • In the process, they use and combine techniques from a large number of subject areas including statistics, data engineering and pattern recognition
• Our collaborative projects are wide and varied
  • Earth Science
  • Neuroscience
  • Genomics
  • Healthcare
  • …

We develop systems that help scientists accomplish their data science more **effectively** and more **efficiently**

This seminar will be based on our experience in enabling these data science cases
A DATA-INTENSIVE COMPUTATIONAL SCIENTIST’S PROBLEM

Traditional approach:
• “see” the instrument-generated data and decide on experiment

Current Problem:
• Too much data to “see”
• Must analyze it directly with big machines

Summary:
- Data analysis is the new bottle-neck limiting physics
- Real-time analysis with super-computing is becoming important also for a few more iterations of Moore’s cycle.
- Super-computers, remotely controlled with web-applications, is a very promising tool for big-data methods to enter mineral physics.
- Future developments may (must?) evolve away from super computers to highly parallelized (GPU’s) local computers and/or cloud computing.
The “big data” problem in data science is not always about storage of exabytes of data and computational grids with petaflop performance.
THE KNOWLEDGE-GAP WE ADDRESS TODAY

• We will not try to define “big data”
• Instead, we discuss a number of “data situations” and technologies
• Data scientists can encounter the “bigness barrier” much before they approach these huge numbers
• We contend that the knowledge gap is mostly related to:
  • Providing the scientists and information analysts with too much of too complex information
  • Challenging the data engineers about providing more analysis than they are used to or integrating data they are not used to
  • Managing the economics of information
  • Figuring out what technology to align with for “my specific problem space”
Angle 1: A key problem with a lot of real world datasets is sparsity. The higher the dimensionality of the data, it is harder to build statistical models on a small dataset.

Angle 2: Some problems are just naturally large, and most current algorithms are inefficient, and many serious computations need a cluster of machines.

Angle 3: It is often the case that algorithms that work on a relatively small amount of data do very well with larger data. So one can research on a small sample, and then on larger data on analytical software (assuming it can support a ton of data) to construct models.

Angle 4: There are whole classes of problems do not work on small datasets, and need extremely large unlabeled sets of data to draw statistical conclusions on their own by minimizing energy functions under constraints.

Angle 5: A lot of data are collected about user behavior. To create recommendation engines and behavior prediction methods, we need a sufficient data set to construct features that are adequate to predict the behavior of every user.

Dimitri Belenco, Google
WHAT HAPPENS WHEN

• A standard analysis program receives more data than it can handle?

• Simple example
  • Edge Detection in images with Canny’s algorithm

• What will happen if this standard algorithm is given an image that is 1 TB in size?

```python
import Image
import numpy
import ndi
import scipy

sigma = 1.4
f = 'lena_std.jpg'
img = Image.open(f).convert('L')
imgdata = numpy.array(img, dtype = float)
G = ndi.filters.gaussian_filter(imgdata, sigma)
sobelout = Image.new('L', img.size)
gradx = numpy.array(sobelout, dtype = float)
grady = numpy.array(sobelout, dtype = float)
sobel_x = [[-1,0,1],[ -2,0,2],[ -1,0,1]]
sobel_y = [[-1,-2,-1],[ 0,0,0],[ 1,2,1]]
width = img.size[1]
height = img.size[0]
for x in range(1, width-1):
    for y in range(1, height-1):
        px = (sobel_x[0][0] * G[x-1][y-1]) + (sobel_x[0][1] * G[x][y-1]) + (sobel_x[0][2] * G[x+1][y-1]) + (sobel_x[1][0] * G[x-1][y]) + (sobel_x[1][1] * G[x][y]) + (sobel_x[1][2] * G[x+1][y]) + (sobel_x[2][0] * G[x-1][y+1]) + (sobel_x[2][1] * G[x][y+1]) + (sobel_x[2][2] * G[x+1][y+1])
        py = (sobel_y[0][0] * G[x-1][y-1]) + (sobel_y[0][1] * G[x][y-1]) + (sobel_y[0][2] * G[x+1][y-1]) + (sobel_y[1][0] * G[x-1][y]) + (sobel_y[1][1] * G[x][y]) + (sobel_y[1][2] * G[x+1][y]) + (sobel_y[2][0] * G[x-1][y+1]) + (sobel_y[2][1] * G[x][y+1]) + (sobel_y[2][2] * G[x+1][y+1])
        gradx[x][y] = px
        grady[x][y] = py
sobeloutmag = scipy.hypot(gradx, grady)
sobeloutdir = scipy.arctan2(grady, gradx)
```
WHAT HAPPENS WHEN

• A database engineer who knows SQL very well is asked to perform potentially complex temporal analysis on data:
  • Find time-partitions in data signifying episodes
  • relationship between HR variability and GSR
WHAT HAPPENS WHEN

• A biologist in a company wants to know which antibodies (from a database) have actually worked (as per publications) in experiments related to their effects on proteins of a specific family (as per a protein classification scheme)

• Integration between relational, textual and hierarchical data

• Direct analogs in commercial world

Both the monoclonal anti-Pdx1 antibodies F6 and F109 were shown to recognize the GST-Pdx1 antigen by western blotting (Figure 1A, lane 3, and data not shown). In addition, the antibodies stained a band of 39 kDa, corresponding to the Pdx1 protein, in nuclear extracts from the SV40 large T-antigen transformed mouse insulinoma βTC6 cell line (Figure 1B, lane 2 and 4), but not from the αTC1.9 glucagonoma mouse cell line (Figure 1B, lane 1 and 3). This again reflects the expected expression of Pdx1 in these two islet tumor lines. This Western-blot staining specificity of F6 and F109 was confirmed by pre-absorption studies, where only pre-incubation with GST-Pdx1 fusion protein and not GST-Nkx6.1 was able to abolish the appearance of the 39 kDa band (data not shown).

The purified monoclonal mouse anti-Pdx1 antibodies F6 and F109 were further characterized by staining of frozen sections of adult mouse pancreas and fetal pancreas. Both antibodies were demonstrated to specifically stain the nuclei of a sub-population of cells in the endocrine islets of Langerhans, as expected for a Pdx1 reactive antibody (Figure 1, C and D). Specificity was demonstrated as the staining of the cell nuclei was eliminated by pre-absorption of the antibodies with the GST-Pdx1 fusion protein (Figure 1, F and H), but not by pre-absorption with an irrelevant GST fusion protein (Figure 1, G and H). Furthermore double staining of adult pancreas with a rabbit anti-Pdx1 antiserum (1858.5) demonstrated a near complete overlap of the nuclei recognized by the two antibodies (Figure 2A–C and D–F). On close
THE V-BASED CHARACTERIZATION OF “BIG DATA”

• **Volume**
  • The amount of data

• **Velocity**
  • The speed of data going in and out

• **Variety**
  • The range of data types & sources

• **Valence**
  • The complex inter-relatedness of data

• **Veracity**
  • The degree of uncertainty in the data

• **Variability**
  • Many options or variable interpretations confound analysis
The “big data” problem can be thought of as different combinations of data characteristics and analytical needs problems.

**Big Volume – Little Analytics**
- Well addressed by database and data warehouse crowd who are pretty good at SQL analytics on
  - Hundreds of nodes and Petabytes of data

**Big Data – Big Analytics**
- Complex math operations (machine learning, clustering, trend detection, ….)
  - the world of the “quants” and data miners
  - Mostly specified as linear algebra on matrix data
- A dozen or so common ‘inner loops’
  - Matrix multiplication, QR decomposition, SVD decomposition, Linear regression, …
• Big Velocity Data
  • Big pattern - little state (electronic trading)
    • Find the occurrence of ‘IBM down’ followed within 100 msec by a ‘MSFT down’
    • Complex event processing (CEP) is focused on this problem
      • Patterns in a fire hose
  • Big state - little pattern
    • For every security, assemble my real-time global position
    • And alert me if my exposure is greater than X
    • Looks like high performance OLTP where we want to update a database at very high speed
  • Big state – big pattern?
Big Variety

- A typical enterprise has many (sometimes thousands of) operational systems
  - Only a few get into a structured, managed database
  - What about the rest?
- For the remaining data
  - One has to look at and be scalable to 1000s of sites
  - Deal with incomplete, conflicting, and incorrect data
- Be incremental because the task is never done
BUT WE WILL START SLOW

THREE (SIMPLIFIED) DATA SCIENCE STORIES

1. A Social Scientist exploring Crime Patterns
2. A Biologist exploring Genetic Variations
3. A Data Discoverer trying to Formulate a Hypothesis

Some Data Computing issues along the way
CASE 1. ANALYZING CRIME DATA

• The (ultra-simplistic) science question
  • Are the more crimes in areas close to places that sell alcohol in the San Diego County?
  • What are dominant types of crimes close to these places?
  • Is there any correlation between the density of alcohol selling places and the nature of the crimes committed in these regions?

• The process
  • Get crime data for San Diego County as completely as possible
  • Get data for alcohol selling locations
  • Perform the necessary analysis by queries and statistics
ASSEMBLING A DATA SET FOR CASE 1

Data and Resources

**SANDAG Crime Incidents 2007**
No description for this resource

**SANDAG Crime Incidents 2008**
No description for this resource

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>activityType</td>
<td>AGENCY</td>
<td>activityDate</td>
<td>LEGEND</td>
<td>Charge_Description</td>
<td>BLOCK_ADDRESS</td>
<td>City_Name</td>
<td>ZipCode</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>CRIME CASE</td>
<td>Carlsbad Police, CA</td>
<td>1/1/2008 00:00</td>
<td>THEFT/LARCENCY</td>
<td>PETTY THEFT</td>
<td>2500 BLOCK ORION WAY</td>
<td>CARLSBAD</td>
<td>92008</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>CRIME CASE</td>
<td>Chula Vista Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>1200 BLOCK AGUIRRE DRIVE</td>
<td>CHULA VISTA</td>
<td>91910</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>CRIME CASE</td>
<td>Chula Vista Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>2400 BLOCK GOLFCREST LOOP</td>
<td>CHULA VISTA</td>
<td>91915</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CRIME CASE</td>
<td>Chula Vista Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>2500 BLOCK CATAMARAN WAY</td>
<td>CHULA VISTA</td>
<td>91914</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>CRIME CASE</td>
<td>Chula Vista Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>400 BLOCK J STREET</td>
<td>CHULA VISTA</td>
<td>91910</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CRIME CASE</td>
<td>El Cajon Police, CA</td>
<td>1/1/2008 00:00</td>
<td>ASSAULT</td>
<td>LEWD/LASCIVIOUS ACTS - CHILD UNDEF</td>
<td>1100 BLOCK TEATRO</td>
<td>EL CAJON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CRIME CASE</td>
<td>El Cajon Police, CA</td>
<td>1/1/2008 00:00</td>
<td>SEX CRIMES</td>
<td>OTHER SEX CRIME</td>
<td>1200 BLOCK PEACH AVENUE</td>
<td>EL CAJON</td>
<td>92021</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CRIME CASE</td>
<td>El Cajon Police, CA</td>
<td>1/1/2008 00:00</td>
<td>SEX CRIMES</td>
<td>OTHER SEX CRIME</td>
<td>700 BLOCK GRAVES AVENUE</td>
<td>EL CAJON</td>
<td>92021</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CRIME CASE</td>
<td>La Mesa Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>7600 BLOCK HOMEWOOD PLACE</td>
<td>LA MESA</td>
<td>91941</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>ASSAULT</td>
<td>THREATEN CRIME WITH INTENT TO TERRORIZE</td>
<td>300 BLOCK MAINSAIL ROAD</td>
<td>OCEANSIDE</td>
<td>92054</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>1300 BLOCK SLEEPING INDIAN ROAD</td>
<td>OCEANSIDE</td>
<td>92057</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>1500 BLOCK HAVENWOOD DRIVE</td>
<td>OCEANSIDE</td>
<td>92056</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>1600 BLOCK HICKS STREET</td>
<td>OCEANSIDE</td>
<td>92054</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>2400 BLOCK CATALINA CIRCLE</td>
<td>OCEANSIDE</td>
<td>92056</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>2800 BLOCK BRANDERIS DRIVE</td>
<td>OCEANSIDE</td>
<td>92056</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>300 BLOCK CALLE VALECITO</td>
<td>OCEANSIDE</td>
<td>92057</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>4500 BLOCK GOLDFINCH WAY</td>
<td>OCEANSIDE</td>
<td>92057</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>4700 BLOCK LOTTY GROVE DRIVE</td>
<td>OCEANSIDE</td>
<td>92056</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>4700 BLOCK YUMA AVENUE</td>
<td>OCEANSIDE</td>
<td>92057</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>500 BLOCK GREENBRIER DRIVE</td>
<td>OCEANSIDE</td>
<td>92054</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>CRIME CASE</td>
<td>Oceanside Police, CA</td>
<td>1/1/2008 00:00</td>
<td>FRAUD</td>
<td>FRAUD</td>
<td>5100 BLOCK NORTH RIVER ROAD</td>
<td>OCEANSIDE</td>
<td>92057</td>
<td></td>
</tr>
</tbody>
</table>

Let’s try to do it the old-fashioned way
A CLOSER LOOK

• What does this assembly process involve?
  • This is a union operation over multiple sets of records (tuples)
  • It involves checking that the records from different sets are of the same kind
    • For example, they are not records with different number of fields
  • The union operation includes elimination of duplicates
• Questions
  • What would happen if
    • We need to perform the union over many more data sets?
    • The size of each data set are much larger?
• It may not be possible for a client-side tool to cope with this data size
  • Excel 2010’s limit: 1048576 rows by 16384 columns – 16.8k from the limit

Is this a “Big Data” Problem? Well, Not Really!
WHY IS MY TOOL FALLING SHORT?

- The tool (software) fails because some data property (volume, complexity, density, …) has changed
- The tool is not designed to effectively utilize our computing platform
  - Processor architecture and programming support
  - Memory
  - I/O bandwidth
  - Disk type and capacity
  for the specific data manipulation and data computing tasks we need
- Should we
  - Get a bigger/faster/newer version of the tool?
  - Use a different tool?
  - Just program it ourselves?

Information Economics Question
CAN’T WE USE A DBMS? (1ST VISIT)

• A DBMS will indeed solve most of these problems
  • While there are many kinds of DBMSs, for now we focus on relational databases
• But many science users do not prefer to use a (relational) DBMS
  • The cost of learning the tools (data definition and data loading and query) doesn’t seem worth it
  • They do not offer good visualization/plotting tools
  • I can handle my data volumes with my programming language
  • They do not support our data types (arrays, spatial, text, etc.)
  • They do not support our access patterns (spatial, temporal, etc.)
  • We tried them but they were too slow
  • We tried them but once we loaded our data we could no longer manipulate and analyze the data using our standard application programs
  • They require an expensive guru (database administrator) to use

From “Scientific data management in the coming decade”
-- Jim Gray et al, 2005
CAN WE JUST PROGRAM IT OURSELVES?

- Back to the example case 1
- We will use Python to perform the same tasks and some more
- We will stay within the boundary of our hardware
FROM SINGLE MACHINE TO MULTI MACHINE
MapReduce is on its way out. But we shouldn’t measure its importance in the number of bytes it crunches, but the fundamental shift in data processing architectures it helped popularise.

This morning, at their I/O Conference, Google revealed that they’re not using Map-Reduce to process data internally at all any more.
DISTRIBUTED FILE SYSTEM

- A file system that is designed to operate on multiple networked computers and partition and replicate blocks of data files over storage available in multiple machines with
  - Location transparency
  - Replication transparency
  - Metadata management
  - Failure transparency
  - Scalability
- Normally a Distributed File System is part of an operating system like NFS, Microsoft’s DFS,…
HADOOP FILE SYSTEM (HDFS)

- distributed file system designed to run on thousands of commodity hardware. ... HDFS is highly fault-tolerant ... HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS enable(s) streaming access to file system data.

- Terabyte sized data across thousands of machines
- Portability across OSs
- Moving computation to data
- Designed for Data Disk Failure
- Strong replication management and re-replication

Others: Ceph parallel file system
MAP-REDUCE AS A PROGRAMMING MODEL

LISP programming language

(map 'length '(() (a) (ab) (abc))) \(\rightarrow\) (0 1 2 3)
(reduce #'+ '(0 1 2 3)) \(\rightarrow\) 6
MAP-REDUCE AS A PROGRAMMING MODEL

- **Map:**
  - reads a collection of values from one or more input sources
  - Invokes a user-defined function Mapper on each element, independently and in parallel
    - Emits key-value pairs
- **Shuffle:**
  - takes the key/value pairs emitted by Mappers and groups together all the key/value pairs with the same key
  - outputs each distinct key and a stream of all the values with that key
- **Reduce:**
  - takes this key-grouped data and invokes a user-defined function, the Reducer on each distinct key-and-values group, independently and in parallel
    - Each Reducer invocation is passed a key and an iterator over all the values associated with that key, and emits values to associate with the input key
**Map Reduce in a Nutshell**

**Map** \((k_1, v_1) \rightarrow \text{list}(k_2, v_2)\)
- Processes one input key/value pair
- Produces a set of intermediate key/value pairs

**Reduce** \((k_2, \text{list}(v_2)) \rightarrow \text{list}(k_3, v_3)\)
- Combines intermediate values for one particular key
- Produces a set of merged output values (usually one)
EXECUTION OF A MAP-REDUCE PROGRAM

- User specifies
  - $M$: no. of map tasks
  - $R$: no. of reduce tasks

- Map Phase
  - input is partitioned into $M$ splits
  - map tasks are distributed across multiple machines

- Reduce Phase
  - reduce tasks are distributed across multiple machines
  - intermediate keys are partitioned (using partitioning function) to be processed by desired reduce task
OK, WHAT IS THE BIG DEAL?

- Many problems from diverse algorithms in Computer Science (and therefore applications making use of these algorithms) can be framed as map-reduce algorithms.

- Social Science Example:
  - Find the common interests of “communities” from amongst 40-50 year olds on Facebook who are interested in music.
    - Could be any other network (e.g., protein-protein interaction network).
  - Central problem – find communities, i.e., find **locally dense subgraphs**.

- There is a “neighborhood similarity” based algorithm that puts nodes with similar neighborhoods (overlapping neighbors) in the same community.
  - This problem has a map-reduce formulation.
  - Group neighbors into c-subsets of size s.
    - Call them “shingles”.
BACK TO COMPUTATION – A CLOSER LOOK
A MUCH SIMPLER EXAMPLE

• Assume two TSV files
  • Users(id, email, language, location)
  • Transactions(transaction-id, product-id, user-id, purchase-amount, item-description)

• Task
  • Find the number of unique locations in which each product has been sold

• As part of this, we need to compute a \textit{join operation} between the two input data sets
  • Users.id = Transactions.user-id
EXAMPLE

Grouping by p’s and counting

INPUTS
- t1, p3, u1, $300, sweater
- t2, p1, u2, $100, chicken
- t3, p1, u1, $100, chicken
- t4, p2, u2, $10, banana
- t5, p4, u4, $9, apple

MAP
- u1, US
- u2, GB
- u3, CA
- u4, CA

REDUCE
- u1, p3
- u1, p1
- u2, p2
- u2, GB
- u4, p4
- u4, CA
- u3, CA

OUTPUT
- p3, US
- p1, US
- p1, GB
- p2, GB
- p4, CA

Partition, sort, group
protected static void runFirstJob(Path transactions, Path users, Path output, Configuration conf) throws Exception {
    Job job = new Job(conf);
    job.setJarByClass(RawMapreduce.class);
    job.setJobName("Raw Mapreduce Step 1");
    job.setPartitionerClass(SecondarySort.SSPartitioner.class);
    job.setGroupingComparatorClass(SecondarySort.SSGroupComparator.class);
    job.setSortComparatorClass(SecondarySort.SSSortComparator.class);

    job.setReducerClass(JoinReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(Text.class);
    job.setOutputFormatClass(SequenceFileOutputFormat.class);

    MultipleInputs.addInputPath(job, transactions, TextInputFormat.class, TransactionMapper.class);
    MultipleInputs.addInputPath(job, users, TextInputFormat.class, UserMapper.class);

    job.setMapOutputKeyClass(TextTuple.class);
    job.setMapOutputValueClass(TextTuple.class);
    FileOutputFormat.setOutputPath(job, output);

    if (job.waitForCompletion(true)) return;
    else throw new Exception("First Job Failed");
}
public class UserMapper extends Mapper<LongWritable, Text, TextTuple, TextTuple> {
    TextTuple outKey = new TextTuple();
    TextTuple outValue = new TextTuple();
    String sortChar = "a";
}

public class TransactionMapper extends Mapper<LongWritable, Text, TextTuple, TextTuple> {
    TextTuple outKey = new TextTuple();
    TextTuple outValue = new TextTuple();

    @Override
    public void map(LongWritable key, Text value, Context context) throws java.io.IOException, InterruptedException {
        String[] record = value.toString().split("\t");
        String productId = record[1];
        String uid = record[2];
        outKey.set(uid, "b");
        outValue.set("product", productId);
        context.write(outKey, outValue);
    }
}
public class JoinReducer extends Reducer<TextTuple, TextTuple, Text, Text> {

    Text location = new Text("UNKNOWN");
    @Override
    public void reduce(TextTuple key, Iterable<TextTuple> values, Context context) throws java.io.IOException, InterruptedException {
        for (TextTuple value: values) {
            if (value.left.toString().equals("location")) {
                location = new Text(value.right);
                continue;
            }

            Text productId = value.right;
            context.write(productId, location);
        }
    }
}
public class SecondStage {

public static class SecondMapper extends Mapper<Text, Text, TextTuple, Text> {

    TextTuple outputKey = new TextTuple();

    @Override
    public void map(Text key, Text value, Mapper.Context context)
    throws java.io.IOException, InterruptedException {
        outputKey.set(key, value);
        context.write(outputKey, value);
    }
}

public static class SecondReducer extends Reducer<TextTuple, Text, Text, LongWritable> {

    LongWritable valueOut = new LongWritable();

    @Override
    public void reduce(TextTuple product, Iterable<Text> locations, Context context)
    throws java.io.IOException, InterruptedException {
        String previous = null;
        long totalLocations = 0;
        for (Text location: locations) {
            if (previous == null || !location.toString().equals(previous)) {
                totalLocations += 1;
                previous = location.toString();
            }
        }
        valueOut.set(totalLocations);
        context.write(product.left, valueOut);
    }
}
SCHEDULING

• Each Job is broken into tasks
  • Map tasks work on fractions of the input dataset, as defined by the underlying distributed file system
  • Reduce tasks work on intermediate inputs and write back to the distributed file system
• The number of tasks may exceed the number of available machines in a cluster
  • The scheduler takes care of maintaining something similar to a queue of pending tasks to be assigned to machines with available resources
• Jobs to be executed in a cluster requires scheduling as well
  • Different users may submit jobs
  • Jobs may be of various complexity
  • Fairness is generally a requirement
SCHEDULING

• **Dealing with stragglers**
  • Job execution time depends on the slowest map and reduce tasks
  • Speculative execution can help with slow machines
    • But data locality may be at stake

• **Dealing with skew in the distribution of values**
  • e.g., temperature readings from sensors
  • In this case, scheduling cannot help
  • It is possible to work on customized partitioning and sampling to solve such issues
TOO PAINFUL?

• Yes, if we are only doing something a database could do
• Assume two TSV files
  • Users(id, email, language, location)
  • Transactions(transaction-id, product-id, user_id, purchase_amount, item_description)
• Task
  • Find the number of unique locations in which each product has been sold
• Simple SQL Query:
  • SELECT t.product_id, count(u.location)
  • FROM Users u, Transactions t
  • WHERE u.id = t.user_id GROUP BY t.product_id

For Hadoop, Higher Level Data Operations are provided through Pig and HIVE
WHO USES PIG?

• An unexpected(?) player
• Pig in Natural Language Processing (courtesy neofone.de)
NAMED ENTITY RECOGNITION

• “Four years after becoming part of Disney, Marvel has emerged as a creative juggernaut for the company as we continue to introduce its rich universe of characters into a variety of businesses including movies, television, theme parks, and consumer products” – Disney Annual Report, 2013

• Which of the several possible entities does a surface term refer to?
  • Computation of 4 probabilities
    • P(entity), P(surface term), P(surface term | entity), P(entity | surface term)
    • P(entity | surface form) = count(entity, surface form) / count(surface form)
    • Estimate using the number of Wikipedia links
FROM NER TO PIG LATIN

parsed = LOAD 'enwiki-20111207-pages-articles.xml',
    USING pignlproc.storage.ParsingWikipediaLoader('en')
    AS (title, id, pageUrl, text, redirect, links, headers, paragraphs);

noredirect = FILTER parsed BY redirect IS NOT NULL;

projected = FOREACH noredirect GENERATE title, text, links, paragraphs;

sentences = FOREACH projected GENERATE title, flatten(pignlproc.evaluation.SentencesWithLink( text, links, paragraphs));

stored = FOREACH sentences GENERATE title, sentenceOrder, linkTarget, linkBegin, linkEnd, sentence;

ordered = ORDER stored BY linkTarget ASC, title ASC, sentenceOrder ASC
STORE ordered INTO '$OUTPUT/$LANG/sentences_with_links'

sentences = LOAD '$INPUT/$LANG/sentences_with_links'
    AS (title: chararray, sentenceOrder: int, linkTarget: chararray, linkBegin: int, linkEnd: int, sentence: chararray);

wikiuri_types = LOAD '$INPUT/$LANG/wikiuri_to_types' AS (wikiuri: chararray, typeuri: chararray);

type_names = LOAD '$TYPE_NAMES' AS (typeuri: chararray, typename: chararray);

-- Perform successive joins to find the OpenNLP typename of the linkTarget

joined = JOIN wikiuri_types BY typeuri, type_names BY typeuri USING 'replicated';

joined_projected = FOREACH joined GENERATE wikiuri, typename;

joined2 = JOIN joined_projected BY wikiuri, sentences BY linkTarget;

result = FOREACH joined2 GENERATE title, sentenceOrder, typename, linkBegin, linkEnd, sentence;
THE BENEFIT AND THE BURDEN

• **Benefits**
  • Provides a different way of looking at the problem
  • The data operations and analyses are decomposable and scalable over a cluster of machines
  • Fine grained control over low-level operations and “how” the process would run

• **Burden for the user**
  • Manages the application logic
  • Manages the infrastructure
  • Tunes the parallelism
BEYOND MAP REDUCE

• When **not** to use Hadoop (and its stack)
  • Your analysis problem does not naturally fit the map reduce model
  • Query response time is an important factor
  • Queries are complex and need significant optimization
  • Map and Reduce operations are too simplistic as primitives
  • Need random, interactive access to data
  • Data security and user access security are really important
• More formally:
  • More **complex**, multi-pass analytics (e.g. ML, graph)
  • More **interactive** ad-hoc queries
  • More **real-time** stream processing
  • Native support for **different data models**
DATA SHARING IN MAPREDUCE

**Input**

- HDFS read
- iter. 1
- HDFS write
- iter. 2
- HDFS read
- ...
DATA SHARING IN SPARK

Input

iter. 1

iter. 2

... one-time processing

Input

Distributed memory

query 1

query 2

query 3

...
Key idea: resilient distributed datasets (RDDs)
- Distributed collections of objects that can be cached in memory across cluster
- Manipulated through parallel operators
- Automatically recomputed on failure
- Contains the basic operations such as map, filter, and persist

Specializations
- `PairRDDFunctions` contains operations available only on RDDs of key-value pairs, such as `groupByKey` and `join`
- `DoubleRDDFunctions` contains operations available only on RDDs of Doubles
- `SequenceFileRDDFunctions` contains operations available on RDDs that can be saved as `SequenceFiles`
A SIMPLE EXAMPLE

SparkContext(master, jobName, [sparkHome], [jars])  

```
lines = spark.textFile("hdfs://...")
errors = lines.filter(_.startsWith("ERROR"))
errors.persist()
```

```
HDFS errors
  filter(_.contains("HDFS"))
  map(_.split('t')(3))
  time fields
```

```
Memory_only, memory_and_disk, memory_serialized, memory_and-disk_2
```

```
errors.filter(_.contains("HDFS"))
  .map(_.split('t')(3))
  .collect()
```

```
errors.filter(_.contains("MySQL")).count()
```
JavaSparkContext sc = new JavaSparkContext(args[0], "JavaKMeans" 
    System.getenv("SPARK_HOME"), JavaSparkContext.jarOfClass(JavaKMeans.class));

JavaRDD<String> lines = sc.textFile(args[1]); 
JavaRDD<double[]> points = lines.map(new ParsePoint());

KMeansModel model = KMeans.train(points.rdd(), k, iterations, runs);

System.out.println("Cluster centers:"); 
for (double[] center : model.clusterCenters()) {
    System.out.println(" " + Arrays.toString(center));
}

double cost = model.computeCost(points.rdd());
System.out.println("Cost: " + cost);
MORE RDD OPERATORS

- map
- filter
- groupBy
- sort
- union
- join
- leftOuterJoin
- rightOuterJoin
- reduce
- count
- fold
- reduceByKey
- groupByKey
- cogroup
- cross
- zip
- sample
- take
- first
- partitionBy
- mapWith
- pipe
- save
...
BDAS – THE BERKELEY DATA STACK

Compatible with Open Source Ecosystem

- *Use* existing interfaces whenever possible

Accept inputs from Kafka, Flume, Twitter, TCP Sockets, ...

Support Hive API

Support HDFS API, S3 API, and Hive metadata
FEATURES OF A DATA ANALYTICS PLATFORM

Winter Corporation Report

- **Architecture**
  - Suitable storage architecture
  - Shared nothing parallelism
  - Cluster size and elasticity
  - Flexible storage model
  - Both bulk load and trickle feed

- **Performance**
  - Extensive data compression and data encoding
  - Read-optimized storage
  - Highly parallel operation
  - Precomputing “hot spot” data
  - Automatic physical database design

- **General Useful and Noteworthy Features for Large-Scale Use**
  - Export-import
  - Backup/restore
  - Workload analyzer
  - Workload management
  - Role-based security

- **Extensions for Advanced Analytics**
  - Query Language extensions
  - Built-in functions
  - User-defined extensions
  - Flexibility in accessing and analyzing all data (structured, semistructured, or unstructured)
CASE 2. EXPLORING GENOMIC VARIATIONS

- The data set (courtesy Scripps Translational Science Institute)
  - Subjects who have lived for a long time with no major diseases have been genotyped
  - Data is the output of a processing pipeline
    - Changing the pipeline will change the schema
      - SG-ADVISER’s output has gone from 50+ to 80+ columns
      - Annotations are performed once and are not updated

- Exploratory science questions
  - Show high-quality genomic variations for subjects with allele frequency = 0.1
  - Which variations related to aging?
  - Execute Madsen and Browning statistical test on individuals returned by query above

- The process
  - Load the data from the variation computation output to a proper DBMS
  - Add any additional data to the database
  - Perform the necessary analysis by queries and statistics
ROW STORE AND COLUMN STORE

- In row store data are stored in the disk tuple by tuple.
- Where in column store data are stored in the disk column by column
## ROW STORE AND COLUMN STORE

<table>
<thead>
<tr>
<th>Row Store</th>
<th>Column Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>(+) Easy to add/modify a record</td>
<td>(+) Only need to read in relevant data</td>
</tr>
<tr>
<td>(-) Might read in unnecessary data</td>
<td>(-) Tuple writes require multiple accesses</td>
</tr>
</tbody>
</table>

- So column stores are suitable for read-mostly, read-intensive, large data repositories
Most of the queries does not process all the attributes of a particular relation.

For example the query

```
Select s.name and s.address
From SUBJECTS as c
Where S.region=San Diego;
```

Only process three attributes of the relation SUBJECTS. But the customer relation can have more than three attributes.

Column-stores are more I/O efficient for read-only queries as they read, only those attributes which are accessed by a query.
ROW STORES

- Row based tables have advantages in the following circumstances:
  - The application needs to only process a single record at one time (many selects and/or updates of single records).
  - The application typically needs to access a complete record (or row).
  - The columns contain mainly distinct values so that the compression rate would be low.
  - Neither aggregations nor fast searching are required.
  - The table has a small number of rows (e.g. configuration tables).
COLUMN STORES

- Column-based tables have advantages in the following circumstances:
  - Calculations are typically executed on single or a few columns only.
  - The table is searched based on values of a few columns.
  - The table has a large number of columns, and more can be added.
  - The table has a large number of rows and columnar operations are required (aggregate, scan etc.)
  - High compression rates can be achieved because the majority of the columns contain only few distinct values (compared to number of rows).
WHY COLUMN STORES?

- Can be significantly faster than row stores for some applications
  - Fetch only required columns for a query
  - Better cache effects
  - Better compression (similar attribute values within a column)
- But can be slower for other applications
  - OLTP with many row inserts, …
  - Large-sized updates
- Long war between the column store and row store camps :-)
COLUMNS STORES – DATA MODEL

- Construct EMP(name, age, salary) from EMP1 and EMP3 using join index on EMP3
COMPRESSION

- Trades I/O for CPU
  - Increased column-store opportunities:
  - Higher data value locality in column stores
  - Techniques such as run length encoding far more useful
- Schemes
  - Null Suppression
  - Dictionary encoding
  - Run Length encoding
  - Bit-Vector encoding
  - Heavyweight schemes
INFORMATION ECONOMICS QUESTIONS

• How much of the buzz around column-stores is marketing hype?
  • Do you really need to buy Sybase IQ or Vertica?
• Can we get column-store performance from a row-store?
• Analytics
  • What is in-database analytics really worth?
• Native analyses
  • Statistics
  • Time-series
  • Standard data mining operations
  • Graph computation
A QUICK DIVERSION
# Example: Auto completing hashtags

<table>
<thead>
<tr>
<th>Suggestions</th>
</tr>
</thead>
<tbody>
<tr>
<td>#argentina, #arugularocks, #argylesocks</td>
</tr>
<tr>
<td>#argentina, #argylesocks, #argonauts</td>
</tr>
<tr>
<td>#argentina, #argentum, #argentine</td>
</tr>
</tbody>
</table>
GOOGLE’S CLOUD DATAFLOW PLATFORM

Tweets

Read

{#argentina scores!, watching #armenia vs #argentina, my #art project, ...}

ExtractTags

{argentina, armenia, argentina, art, ...}

Count

{argentina->5M, armenia->2M, art->90M, ...}

ExpandPrefixes

{a->(argentina, 5M), a->(armenia, 2M), ... , ar->(argentina, 5M), ar->(armenia, 2M), ...}

Top(3)

{a->[apple, art, argentina], ar->[art, argentina, armenia], ...}

Write

Predictions
GOOGLE’S CLOUD DATAFLOW PLATFORM

```
Pipeline p = Pipeline.create();
p.begin()
    .apply(TextIO.Read.from("gs://..."))
    .apply(ParDo.of(new ExtractTags()))
    .apply(Count.create())
    .apply(ParDo.of(new ExpandPrefixes()))
    .apply(Top.largestPerKey(3))
    .apply(TextIO.Write.to("gs://..."));
p.run();
```
STREAMING DATA WITH GOOGLE'S CLOUD DATAFLOW PLATFORM

Google Cloud Pub/Sub for reads and writes.

Age out old data

#ar* rank

#argyle

#argentinagoal

#armeniarocks

game begins  armenia wins!  time
Streaming Data with Google's Cloud Dataflow Platform

Pipeline p = Pipeline.create();
   p.begin()
      .apply(PubsubI0.Read.from("input_topic"))
      .apply(Bucket.by(SlidingWindows.of(60, MINUTES)))
      .apply(ParDo.of(new ExtractTags()))
      .apply(Count.create())
      .apply(ParDo.of(new ExpandPrefixes()))
      .apply(Top.largestPerKey(3))
      .apply(PubsubI0.Write.to("output_topic"));
   p.run();
STREAMING DATA WITH GOOGLE'S CLOUD DATAFLOW PLATFORM
THE VARIETY PROBLEM – DATA GENRES

- Structured Data
  - Relations – tables
  - Nested relations – HTML tables
- Unstructured Data
  - Text – the content needs to be retrieved/analyzed independent of its structure
  - Images and Videos
- Semi-structured Data
  - XML – direct XML, PowerPoint, Word files, …
  - JSON
  - Graph-structured data – social networks, citation networks, …
  - Array-structured data – images, matrix operations, simulation outputs
- Domain specific Structures
  - HDF, NetCDF – for scientific data
  - -- for physiology time series
  - …
create dataverse LittleTwitterDemo;

create type TweetMessageType as open {
    tweetid: string,
    user: {
        screen-name: string,
        lang: string,
        friends_count: int32,
        statuses_count: int32,
        name: string,
        followers_count: int32
    },
    sender-location: point?,
    send-time: datetime,
    referred-topics: {{ string }},
    message-text: string
};

create dataset TweetMessages(TweetMessageType)
primary key tweetid;

**Highlights:**
- JSON++ based data model
- Rich type support (spatial, temporal, …)
- Records, lists, bags
- **Open vs. closed types**
- External data sets and datafeeds
EX: TWEETMESSAGES DATASET

```
{
    "tweetid": "1023",
    "user": {
        "screen-name": "dflynn24",
        "lang": "en",
        "friends_count": 46,
        "statuses_count": 987,
        "name": "danielle flynn",
        "followers_count": 47
    },
    "sender-location": "40.904177,-72.958996",
    "send-time": "2010-02-21T11:56:02-05:00",
    "referred-topics": { "verizon" },
    "message-text": "i need a #verizon phone like nowwww :(
    }
},
{
    "tweetid": "1024",
    "user": {
        "screen-name": "miriamorous",
        "lang": "en",
        "friends_count": 69,
        "statuses_count": 1068,
        "name": "Miriam Songco",
        "followers_count": 78
    },
    "send-time": "2010-02-21T11:43:08-08:00",
    "referred-topics": { "commercials", "verizon", "att" },
    "message-text": "#verizon & #att #commercials, so competitive"
},
{
    "tweetid": "1025",
    "user": {
        "screen-name": "dj33",
        "lang": "en",
        "friends_count": 96,
        "statuses_count": 1696,
        "name": "Don Jango",
        "followers_count": 22
    },
    "send-time": "2010-02-21T12:38:44-05:00",
    "referred-topics": { "verizon", "att" },
    "message-text": "Chillin at dca waiting for 900am flight to #charlotte and from there to providenciales"
},
{
    "tweetid": "1026",
    "user": {
        "screen-name": "reallyleila",
        "lang": "en",
        "friends_count": 106,
        "statuses_count": 107,
        "name": "Leila Samii",
        "followers_count": 52
    },
    "send-time": "2010-02-21T21:31:57-06:00",
    "referred-topics": { "verizon", "at&t", "iphone" },
    "message-text": "I think a switch from #verizon to #at&t may be in my near future... my smartphone is like a land line compared to the #iphone!"
}]]
```
**ASTERIX QUERY LANGUAGE (AQL)**

- **Ex:** List the user information and tweet message text for Verizon-related Tweets:

```plaintext
for $tweet in dataset TweetMessages
where $tweet.user.screen-name = 'dflynn24'
return {
  "tweeter": $tweet.user,
  "tweet": $tweet.message-text
}
```

**Highlights:**
- Set-similarity matching (\(\sim\) operator)
- Spatial predicates and aggregation

#ASTERIXDB
AQL (CONT.)

- **Ex:** List the topics being Tweeted about, along with their associated Tweet counts, in Verizon-related Tweets:

```
for $tweet in dataset TweetMessages
where some $topic in $tweet.referred-topics
    satisfies contains($topic, "verizon")
for $topic in $tweet.referred-topics
group by $topic with $tweet
return {
    "topic": $topic,
    "count": count($tweet)
}
```
FUZZY JOINS IN AQL

Ex: Find Tweets with similar content:

```plaintext
for $tweet1 in dataset TweetMessages
for $tweet2 in dataset TweetMessages
where $tweet1.tweetid != $tweet2.tweetid
    and $tweet1.message-text ~= $tweet2.message-text
return {
    "tweet1-text": $tweet1.message-text,
    "tweet2-text": $tweet2.message-text
}
```
CONTINUOUS DATA FEEDS (FUTURE)

- Ex: Create “Fast Data” feeds for Tweets and News articles:

```sql
create feed dataset TweetMessages(TweetMessageType)
using TwitterAdapter ("interval"="10")
apply function addHashTagsToTweet
primary key tweetid;

create feed dataset NewsStories(NewsType)
using CNNFeedAdapter ("topic"="politics","interval"="600")
apply function getTaggedNews
primary key storyid;

create index locationIndex on Tweets(sender-location) type rtree;
```

Begin feed TweetMessages;

Begin feed NewsStories;

Highlights:
- Philosophy: “keep everything”
- Data ingestion, not data streams
- Previous queries unchanged

#ASTERIXDB
THE ASTERIX SOFTWARE STACK

AsterixQL

- Asterix Data Mgmt. System
  - Hivesterix
  - Other HLL Compilers

- Algebricks Algebra Layer
- Hadoop M/R Compatibility
- Pregelix
- IMRU

Hyracks Data-parallel Platform
ALGEBRICKS

Set of (data model agnostic) logical operations
Set of physical operations
Rewrite rule framework (logical, physical)
Generally applicable rewrite rules (including parallelism)
Metadata provider API (catalog info for Algebricks)
Mapping of physical operations to Hyracks operators
HYRACKS

• Partitioned-parallel platform for data-intensive computing
• Job = dataflow DAG of operators and connectors
  – Operators consume and produce partitions of data
  – Connectors route (repartition) data between operators

HYRACKS vs. the “competition”
  – Based on time-tested parallel database principles
  – vs. Hadoop: More flexible model and less “pessimistic”
  – vs. Dryad: Supports data as a first-class citizen
HYRACKS: OPERATOR ACTIVITIES

{NC1: cust1.dat}
{NC2: cust2.dat}

Scanner (CUSTOMER) → E1[hash(C_CUSTKEY)]

HashJoin
C_CUSTKEY = O_CUSTKEY

E2[hash(O_CUSTKEY)] → E3

HashGroupby
C_MKTSEGMENT
Agg: count(O_ORDKEY)

E3[hash(C_MKTSEGMENT)] → E4[1:1]

Writer

{NC3: ord1.dat, NC2: ord1.dat}
{NC1: cust2.dat, NC5: ord2.dat}

Scanner (ORDERS)

HashJoin

JoinBuild → JoinProbe

E1

HashGroupby
HashAggregate → OutputGenerator

E3

E4[1:1]

Writer
HYRACKS: RUNTIME TASK GRAPH

Stage 1:
- S11
- S12
- JB1
- JB2
- JB3
- JB4

Stage 2:
- JB1
- JB2
- JB3
- JB4
- S21
- S22

Stage 3:
- HA1
- HA2
- HA3
- OG1
- W1
- OG2
- W2
- OG3
- W3

Legend:
- JB1..4: Join Build
- JP1..4: Join Probe
- HA1..3: Hash Aggregate
- OG1..3: Output Generator

A → B
A's output is hash distributed to B

A → B
B blocks until A completes

A → B
A's output is piped to B directly

A → B
A's output is piped to B and multiple inputs at B are merged into one
The architecture of a typical text search system

- Acquire content
- Build document
- Analyze document
- Index document

Index

Users

- Build query
- Render results
- Run query

Raw Content
HOW APACHE LUCENE MODELS CONTENT

- A Document is the atomic unit of indexing and searching
  - A Document contains Fields
- Fields have a name and a value
  - You have to translate raw content into Fields
    - Examples: Title, author, date, abstract, body, URL, keywords, ...
  - Different documents can have different fields
    - Search a field using name:term, e.g., title:lucene
  - Be indexed or not
    - Indexed fields may or may not be analyzed (i.e., tokenized with an Analyzer)
      - Non-analyzed fields view the entire value as a single token (useful for URLs, paths, dates, social security numbers, ...)
  - Be stored or not
    - Useful for fields that you’d like to display to users
- Optionally store term vectors
  - Like an inverted index on the Field’s terms
  - Useful for highlighting, finding similar documents, categorization
SEARCHING TEXT IN THE LARGE – SOLR CLOUD

Solr: an enterprise search platform that uses Lucene
DISTRIBUTED INDEXING OF TEXT (AND DATA)

- A collection is a **distributed index** defined by:
  - **named configuration** stored in ZooKeeper
  - **number of shards**: documents are distributed across N partitions of the index
  - **document routing strategy**: how documents get assigned to shards
  - **replication factor**: how many copies of each document in the collection
### EXAMPLES OF LUCENE QUERIES

<table>
<thead>
<tr>
<th>Query expression</th>
<th>Document matches if…</th>
</tr>
</thead>
<tbody>
<tr>
<td>java</td>
<td>Contains the term <em>java</em> in the default field</td>
</tr>
<tr>
<td>java junit, java OR junit</td>
<td>Contains the term <em>java</em> or <em>junit</em> or both in the default field (the default operator can be changed to AND)</td>
</tr>
<tr>
<td>+java +junit, java AND junit</td>
<td>Contains both <em>java</em> and <em>junit</em> in the default field</td>
</tr>
<tr>
<td>title:ant</td>
<td>Contains the term <em>ant</em> in the title field</td>
</tr>
<tr>
<td>title:extreme –subject:sports</td>
<td>Contains <em>extreme</em> in the title and not <em>sports</em> in subject</td>
</tr>
<tr>
<td>(agile OR extreme) AND java</td>
<td>Boolean expression matches</td>
</tr>
<tr>
<td>title:&quot;junit in action&quot;</td>
<td>Phrase matches in title</td>
</tr>
<tr>
<td>title:&quot;junit action&quot;~5</td>
<td>Proximity matches (within 5) in title</td>
</tr>
<tr>
<td>java*</td>
<td>Wildcard matches</td>
</tr>
<tr>
<td>java~</td>
<td>Fuzzy matches</td>
</tr>
<tr>
<td>lastmodified:[1/1/09 TO 12/31/09]</td>
<td>Range matches</td>
</tr>
</tbody>
</table>
EXAMPLES

• PubMed is a corpus of biomedical articles that we index and search as part of the Neuroscience Information Framework (NIF) Project

• Solr is used as the underlying searching engine
  • 23,498,941 documents, 126GB on disk
  • search for hippocampus in title or abstract
  • search for hippocampus antidementia drug in title or abstract with title boosted
  • search for captions containing hippocampus slice
  • find journal names where we find immunocytochemistry within 5 words of experiments in the abstract
  • find documents with "Alzheimer's Disease" in the title, and PSEN2 in the methods section
  • when did the term tractography start appearing in the introduction of papers? For which journal?
GRAPH DATA

The Matrix
Title: The Matrix
Release: 1999

Keanu Reeves
Born: 1964

ACTED_IN
Role: Neo

Node category
Edge Type
Edge property
Node property

Node name
Edge (relationship)
QUERYING WITH CYPHER, A GRAPH QUERY LANGUAGE

• Find a small subgraph with 25 edges
  • MATCH a-[r]-b return a, r, b limit 25
QUERYING WITH CYPER, A GRAPH QUERY LANGUAGE

- Find a subgraph of actors and movies released between 1995 and 2012

```cypher
match (m:Movie) where m.released > 1995 and m.released < 2012 with m
match p = a-[:ACTED_IN]-> (m) return p
```
QUERYING WITH CYPHER, A GRAPH QUERY LANGUAGE

- Find a 5-distance network between Keanu Reeves and Carrie-Anne Moss

  MATCH (a:Person {name:"Keanu Reeves"}), (b:Person {name:"Carrie-Anne Moss"})
  MATCH p = (a)-[r*0..5]-(b)
  RETURN p;
SCIGRAPH

- A graph engine developed over neo4j
  - Vocabulary operations
    - Goal: Match free text to concepts
    - Internally represented as node properties (label, synonyms, definition)
    - Augmented by a custom Lucene index forgiving minor variations (punctuation, lemmatization, etc)
  - Graph Operations
    - Simple paths
    - Transitive closure
THE VARIETY PROBLEM – DATA GENRES

- Structured Data
  - Relations – tables
  - Nested relations – HTML tables
- Unstructured Data
  - Text – the content needs to be retrieved/analyzed independent of its structure
  - Images and Videos
- Semi-structured Data
  - XML – direct XML, PowerPoint, Word files, …
  - JSON
  - Graph-structured data – social networks, citation networks, …
  - Array-structured data – images, matrix operations, simulation outputs
- Domain specific Structures
  - HDF, NetCDF – for scientific data
  - -- for physiology time series

‘What happens when you have a data integration problem involving all these data genres?’
CASE 3: MULTI-GENRE INFORMATION INTEGRATION

• How to find a biomedical resource on the web?
  • Resource
    • Anything that is tangible and accessible e.g., a product, a person, an institution, a piece of data, a connection …

• Why is this an information integration problem?
  • Many sources have information about the same kinds of objects
  • Different integration goals (next slide)

• Basic Process
  • Find potential data by automated and human-contributed means
  • Gather and correlate them by mapping
  • Index and search/query over them
RESOURCE FINDING IN BIOMEDICAL SCIENCES

The problem starts here

Where is data about X?
✓ How does Y relate to Z?
✓ Accumulate and Analyze
✓ Compare X and Y
✓ Subscribe to topic T
✓ Recommend Resource
✓ Funding reports
✓ Search and Explore
✓ News

Resource Activity

✓ Resource Promotion
✓ Utilization Search
✓ Cross-Utilization
✓ Experiential Services

Analysis & Science

Data Infrastructure

Data
ARCHITECTURE
THE “INFORMATION VARIETY” PROBLEM

• The data come
  • From too many disparate sources
    • 7000+ neuroscience resources
  • In too many different formats and models
    • Relational, XML, RDF, Text, domain-specific, …
  • Having all too diverse semantics
    • “GRM1”: a string, a gene, a chromosomal region, a list of interesting SNPs in mice?

• There is a massive data integration problem because only integration of data will lead to insight
  • What possible drugs might be repurposed for human inclusion body myopathy (HIBM)?
  • Data about/from the following to be integrated
    • Organisms, diseases, cross-organism anatomy, phenotypes, genes, proteins, interactions, pathways, genomic variations, pharmaceutical compounds, assays and publications, …
Which skeletal structures in the zebrafish develop from the mesenchyme?

Return $x$ where

- ($x \text{ subclassOf} \text{\ 'skeletal element'}$) and
- ($x \text{ develops_from} \text{\ 'ZFA:mesenchyme'}$)

Return $y$ where

- ($y \text{ subclassOf} \text{\ $x$}$)

Query Rewriting

Return $x$ where

- ($x \text{ subclassOf} \text{\ 'skeletal element'}$)
- ($x \text{ develops_from} \text{\ $y$}$)
- ($y \text{ equivalent_to} \text{\ $m$}$)
- ($m \text{ has_ontology} \text{\ ZFA}$)
- ($m \text{ develops_from} \text{\ 'mesenchyme'}$)
Subclasses of “skeletal element” (incl. its equivalenceClasses) in ZFA
http://nif-services-stage.neuinfo.org/ontoquest-lamhdi/rel/subclasses/term/skeletal%20element?level=3

“develops_from”
http://nif-services-stage.neuinfo.org/ontoquest-lamhdi/rel/edge-relation/id/RO_0002202

“equivalenceClass”
http://nif-services-stage.neuinfo.org/ontoquest-lamhdi/rel/children/term/skeletal%20element?level=1
DATA ARE CONNECTED THROUGH THE ONTOLOGY

- What are the top 10 genes you have most information about?
  - Which sources have information about “genes”?
    - Find the ontology term for gene → CHEBI_23367
    - Call
      - http://cm.neuinfo.org:8080/cm_services/column/mapping/ontoterm?ontologyTermId=CHEBI_23367

- For each table and column thus found, call
  - http://cm.neuinfo.org:8080/cm_services/column/valuefreqs?
    - srcNifId=nlx_146253&tableNifId=nlx_146253-1&columnName=transgenic_line

- Now group and rank merge
  - in parallel if needed
  - Get top 10
PUTTING EVERYTHING TOGETHER

The PDSP $K_i$ database is a unique resource in the public domain which provides information on the abilities of drugs to interact with an expanding number of molecular targets. The $K_i$ database serves as a data warehouse for published and internally-derived $K_i$ or affinity, values for a large number of drugs and drug candidates at an expanding number of G-protein coupled receptors, ion channels, transporters and enzymes.

public static final String THC_QUERY = "cannabis thc marijuana";
public void demonstrateFederation() throws Exception {
    final String kiDatabaseId = "nif-0000-01866-1";
    FederationQuery query = FederationQuery.builder(kiDatabaseId, THC_QUERY).get();
    for (Facets facets: searcher.getFacets(query, 10, 0, 1)) {
        // Find all receptor (gene) facets
        if (!facets.getCategory().equals("Receptor")) {
            continue;
        }
        for (Facet facet: facets.getFacets()) {
            // Get grants related to these genes from NCI
            query = FederationQuery.builder("nif-0000-10319-1", "" + facet.getFacet() + """)
                    .facet("Funding Institute", "national cancer institute")
                    .exportType(ExportType.data)
                    .rows(1000).get();
            TableData data = searcher.getTableData(query);

            for (FederationModelData model: data.getResult()) {
                System.out.println(facet.getFacet() + "," + model.get("project_number") + "," + model.get("project_title"));
            }
        }
    }
}

Which marijuana related genes are of interest to NCI? Let’s just use PDSP as an example.
<table>
<thead>
<tr>
<th>Gene</th>
<th>Grant ID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>cannabinoid cb2</td>
<td>3R01CA142115-04S1</td>
<td>Cannabinoid CB2 Agonists for Treatment of Breast Cancer-Induced Bone Pain</td>
</tr>
<tr>
<td>cannabinoid cb2</td>
<td>1R01CA142115-01A1</td>
<td>Cannabinoid CB2 Agonists for Treatment of Breast Cancer-Induced Bone Pain</td>
</tr>
<tr>
<td>sigma</td>
<td>1R01CA163764-01</td>
<td>SIGMA-2/PEPTIDOMIMETIC CONJUGATES TARGET APOPTOSIS IN PANCREATIC CANCER</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC008714-35</td>
<td>Bacterial Functions Involved in Cell Growth Control</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC010632-09</td>
<td>Transcription Regulation in E. coli and H. pylori</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC010632-08</td>
<td>Transcription Regulation in E. coli and H. pylori</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC010632-07</td>
<td>Transcription Regulation in E. coli and H. pylori</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC008714-32</td>
<td>Bacterial Functions Involved in Cell Growth Control</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC008714-34</td>
<td>Bacterial Functions Involved in Cell Growth Control</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC008714-33</td>
<td>Bacterial Functions Involved in Cell Growth Control</td>
</tr>
<tr>
<td>sigma</td>
<td>1R21CA173887-01A1</td>
<td>Nanomicellar Formulation for Synergistic Targeting of Prostate Cancer</td>
</tr>
<tr>
<td>sigma</td>
<td>1F32CA171543-01</td>
<td>Synthesis of Vinblastine Analogues with Improved Physiochemical Properties</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC010378-13</td>
<td>Macromolecular Crystallography Research with Synchrotron Radiation</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC011203-04</td>
<td>Proteolysis and Regulation of Bacterial Cell Growth Control</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC011203-03</td>
<td>Proteolysis and Regulation of Bacterial Cell Growth Control</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC010845-04</td>
<td>p53-induced Regulation of Transcription in the Chromatin Context</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC011203-02</td>
<td>Proteolysis and Regulation of Bacterial Cell Growth Control</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZICBC010517-08</td>
<td>Large Databases of Small Molecules - Drug Development Tool and Public Resource</td>
</tr>
<tr>
<td>sigma</td>
<td>2R56CA107510-06</td>
<td>The role of p53 and 14-3-3 in genomic instability</td>
</tr>
<tr>
<td>sigma</td>
<td>1ZIABC011203-01</td>
<td>Proteolysis and Regulation of Bacterial Cell Growth Control</td>
</tr>
<tr>
<td>gene</td>
<td>projNum</td>
<td>projTitle</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>bptf</td>
<td>1ZIABC005263-28</td>
<td>Eukaryotic Chromatin Structure and Gene Regulation</td>
</tr>
<tr>
<td>bptf</td>
<td>1ZIABC005263-29</td>
<td>Eukaryotic Chromatin Structure and Gene Regulation</td>
</tr>
<tr>
<td>bptf</td>
<td>1ZIABC005263-31</td>
<td>Eukaryotic Chromatin Structure and Gene Regulation</td>
</tr>
<tr>
<td>bptf</td>
<td>1ZIABC005263-30</td>
<td>Eukaryotic Chromatin Structure and Gene Regulation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC011267-03</td>
<td>Preclinical drug development in pancreatic cancer</td>
</tr>
<tr>
<td>itk</td>
<td>1R41CA167907-01</td>
<td>Calibrated Methods for Quantitative PET/CT Imaging</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC009281-26</td>
<td>Receptor Mediated T and B Cell Activation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC010304-14</td>
<td>Biochemical Basis of T Cell Activation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC009281-25</td>
<td>Receptor Mediated T and B Cell Activation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC010304-13</td>
<td>Biochemical Basis of T Cell Activation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC010944-04</td>
<td>Control of the immune response for cancer vaccine development</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC009281-24</td>
<td>Receptor Mediated T and B Cell Activation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC010304-12</td>
<td>Biochemical Basis of T Cell Activation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC010944-03</td>
<td>Control of the immune response for cancer vaccine development</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC010304-11</td>
<td>Biochemical Basis of T Cell Activation</td>
</tr>
<tr>
<td>itk</td>
<td>1ZIABC010944-02</td>
<td>Control of the immune response for cancer vaccine development</td>
</tr>
<tr>
<td>itk</td>
<td>3R01CA112663-10S1</td>
<td>T-bet and Tumor Immunity</td>
</tr>
<tr>
<td>crkl</td>
<td>1ZIASC006892-23</td>
<td>Molecular Biology of Pediatric Tumors</td>
</tr>
<tr>
<td>crkl</td>
<td>1ZIASC006892-24</td>
<td>Molecular Biology of Pediatric Tumors</td>
</tr>
<tr>
<td>crkl</td>
<td>1ZIASC006892-22</td>
<td>Molecular Biology of Pediatric Tumors</td>
</tr>
<tr>
<td>crkl</td>
<td>1ZIASC006892-21</td>
<td>Molecular Biology of Pediatric Tumors</td>
</tr>
<tr>
<td>tetrahydrocannabinol (thc)</td>
<td>3R01CA111196-04S1</td>
<td>MODULATION OF ONCOGENIC AGENTS BY MARIJUANA</td>
</tr>
<tr>
<td>brca1</td>
<td>2R01CA089239-13A1</td>
<td>Analysis of BRCA1 function in DNA Repair</td>
</tr>
<tr>
<td>brca1</td>
<td>7R01CA111436-04</td>
<td>Regulation of BRCA1 Function by Protein Phosphatase 1</td>
</tr>
<tr>
<td>brca1</td>
<td>1R01CA137023-01A1</td>
<td>The Role of BRCA1/BARD1 in Basal-like Breast Cancer</td>
</tr>
<tr>
<td>brca1</td>
<td>4R01CA129440-03</td>
<td>ROLE OF BRCA1/AKT1 PATHWAY IN THE TUMORIGENESIS</td>
</tr>
<tr>
<td>brca1</td>
<td>1R01CA129440-01A2</td>
<td>ROLE OF BRCA1/AKT1 PATHWAY IN THE TUMORIGENESIS</td>
</tr>
<tr>
<td>brca1</td>
<td>7R01CA089239-11</td>
<td>Analysis of BRCA1 Function in DNA Repair</td>
</tr>
<tr>
<td>brca1</td>
<td>1R01CA174904-01</td>
<td>Roles of Chromatin Modification in BRCA1 Dependent DNA Repair</td>
</tr>
<tr>
<td>brca1</td>
<td>1ZIABC010847-06</td>
<td>Gene-specific Mechanisms of BRCA1 transcriptional Control</td>
</tr>
</tbody>
</table>
ISSUES WE DON’T THINK ABOUT

• Number of large concurrent queries/analyses
  • Large number of predicates
  • Large volume of results
  • Analyses with expected time bounds
• Data Loading and movement
• Data Cleaning, Curation and Provenance
• Formulating meaningful exploration and queries and analyses against poorly understood big data
• Ensuring quality of results from queries and analysis
QUESTIONS?