

CS 444: Big Data Systems

Chapter 3. Overview of Big Data Ecosystem

Chase Wu

New Jersey Institute of Technology

Some of the slides were provided through the courtesy of Dr. Ching-Yung Lin
at Columbia University

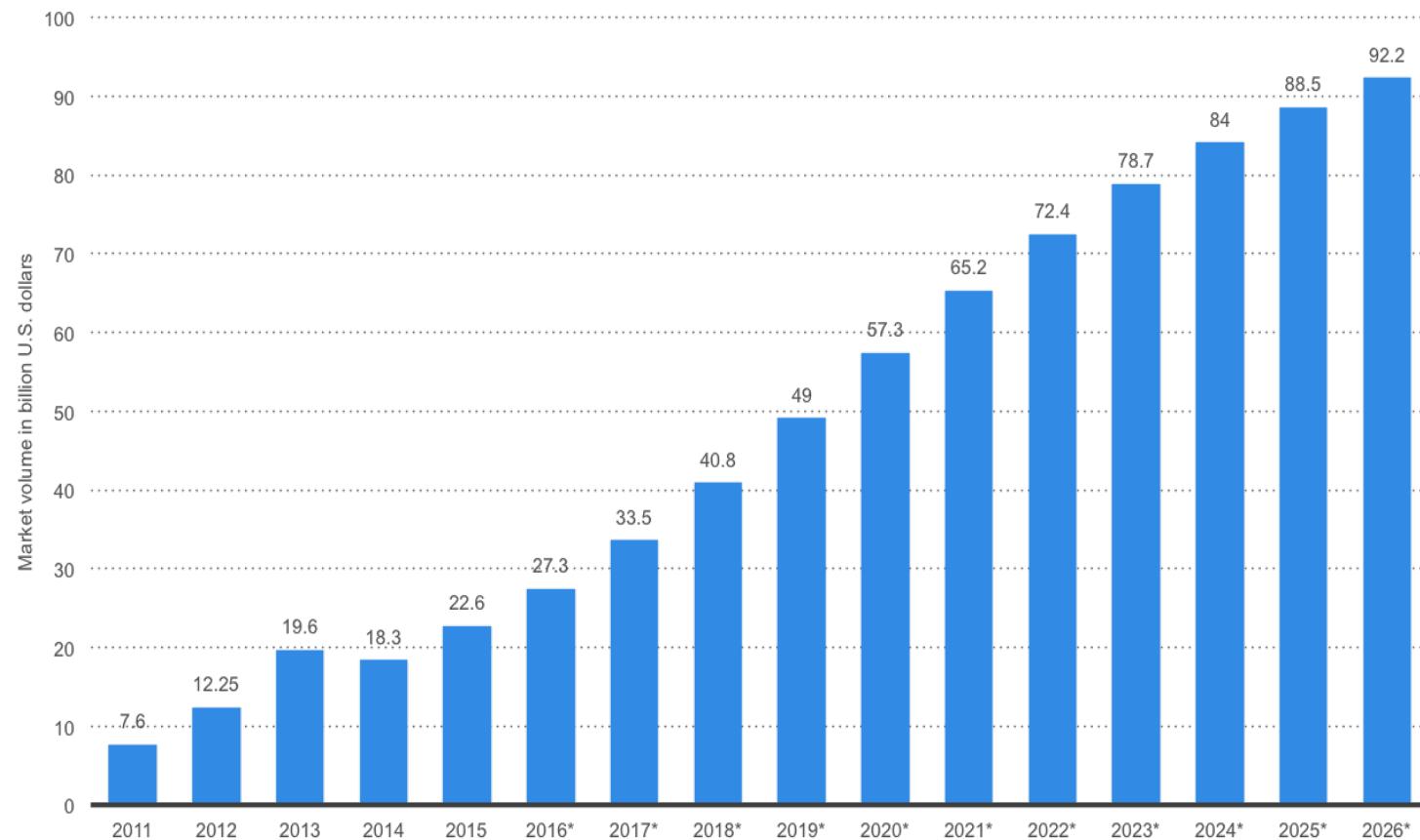
Big Data



Forecast revenue big data market worldwide 2011-2026

Big data market size revenue **forecast** worldwide from 2011 to 2026 (in billion U.S. dollars)

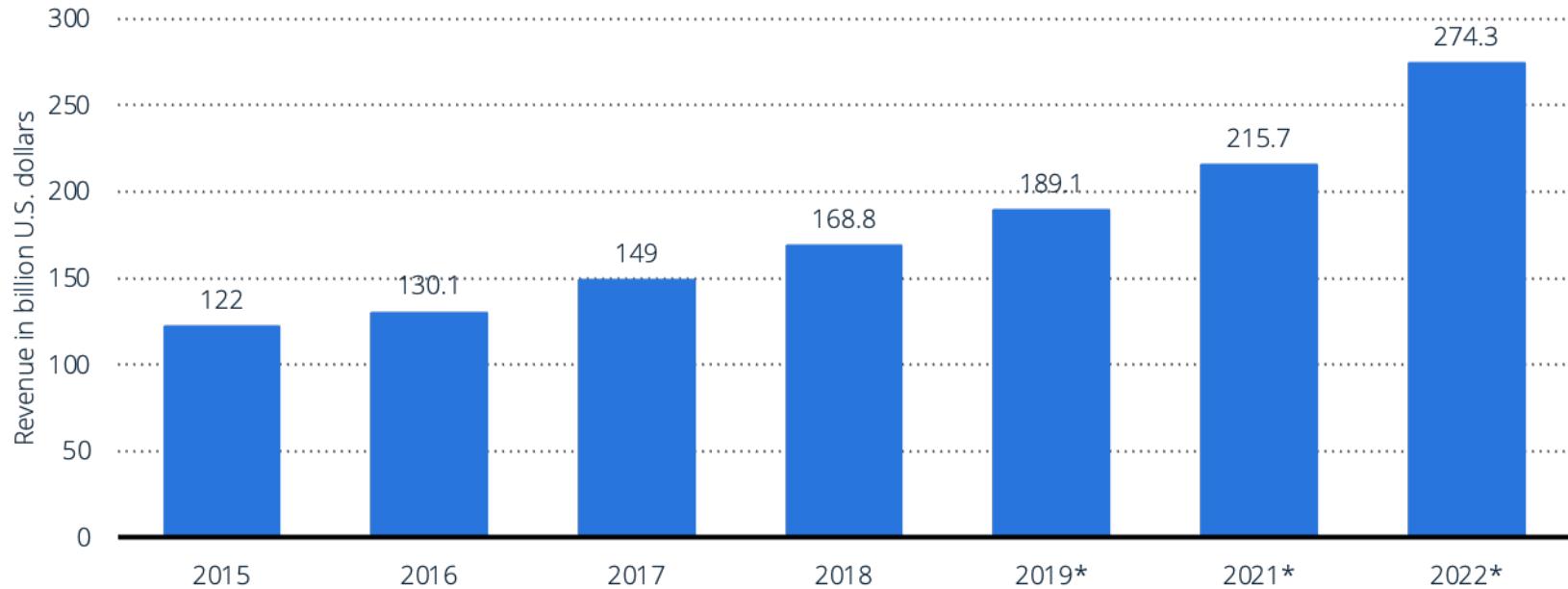
The Big Data market is exploding, not only in terms of marketing hype, but also in real revenue



Note: Worldwide; 2014 to 2016

Source: Wikibon; [ID 254266](#)

Revenue from big data and business analytics worldwide from 2015 to 2022 (in billion U.S. dollars)



2022: ~4 times more than predicted!

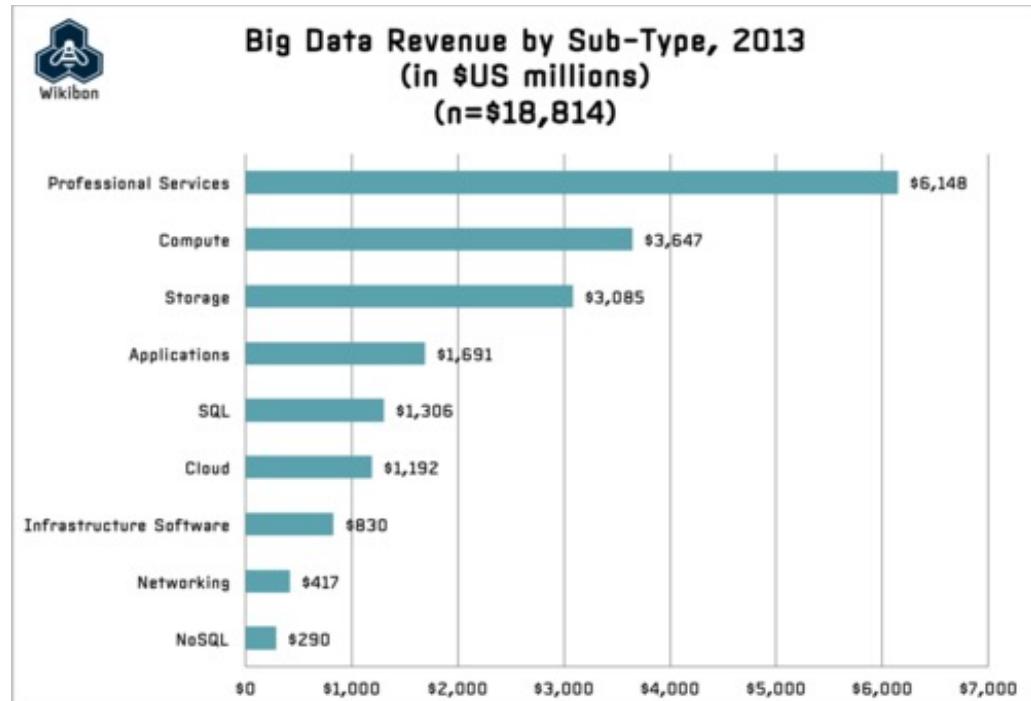
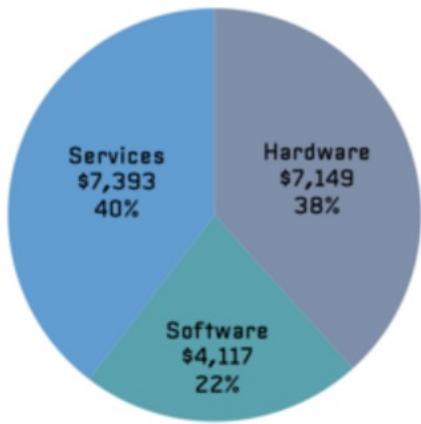
Note(s): Worldwide; 2015 to 2021
Source(s): IDC; [ID 551501](#)

statista

Big Data Revenue By Type



Big Data Revenue by Type, 2013
(in \$US millions)
(n=\$18,814)



5 Key Big Data Use Case Categories – IBM's Perspective



Big Data Exploration

Find, visualize, and understand all big data to improve decision making



Enhanced 360° View of the Customer

Extend existing customer views (MDM, CRM, etc.) by incorporating additional internal and external information sources



Security/Intelligence Extension

Lower risk, detect fraud and monitor cyber security in real-time



Operations Analysis

Analyze a variety of machine data for improved business results



Data Warehouse Augmentation

Integrate big data and data warehouse capabilities to increase operational efficiency

Key Computing Resources for Big Data

- Processing capability: CPU, multi/many-core processor, or node
- Memory
- Storage
- Network

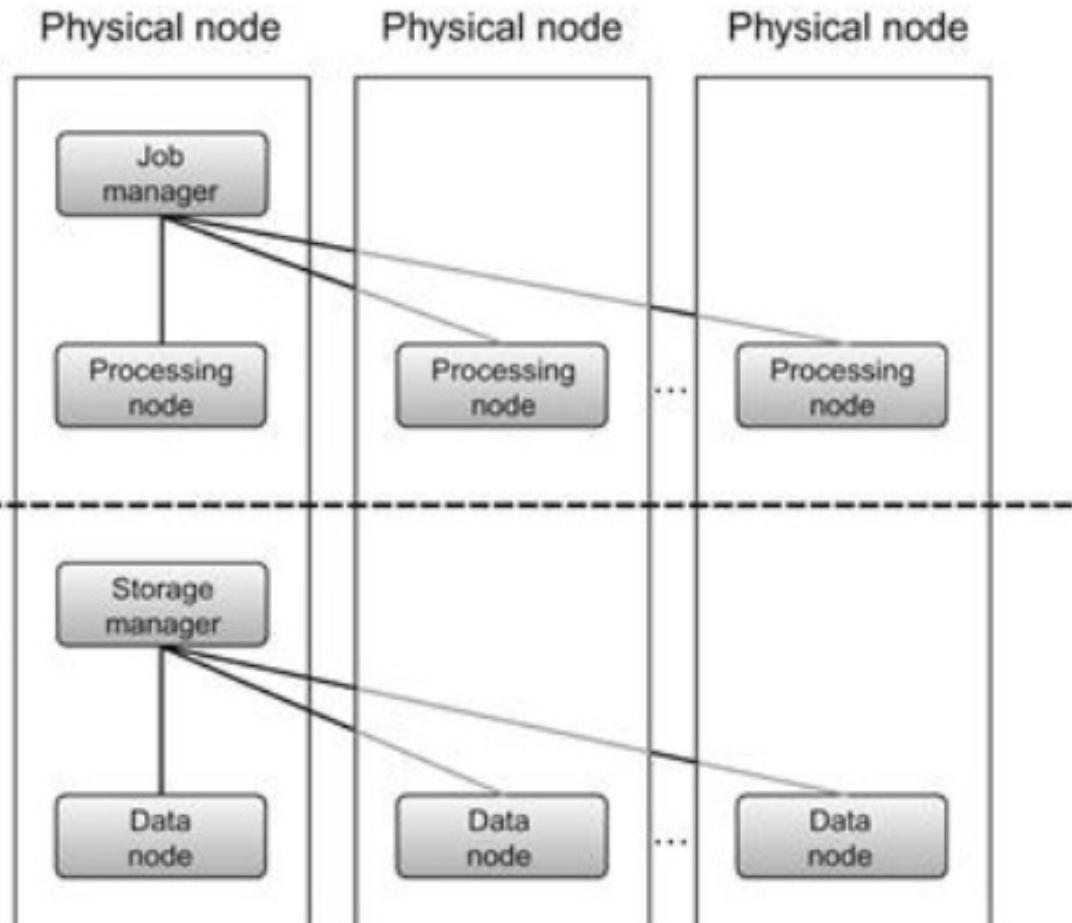
MapReduce, Spark (Computing)

Executive management

Storage/data management

HDFS (Storage)

Name node + Data nodes



"Big Data Analytics", David Loshin, 2013

Techniques towards Big Data

- Massive Parallelism
- Huge Data Volumes Storage
- Data Distribution
- High-Speed Networks
- High-Performance Computing
- Task and Thread Management
- Data Mining and Analytics
- Data Retrieval
- Machine Learning
- Data Visualization

→ Techniques exist for years to decades. Why did Big Data become **hot** now?

Why Big Data now?

- More data are being collected and stored
- Open-source code
- Commodity hardware
- Successful applications of data-driven AI and ML techniques, such as the recent GPTs.

The driving force behind big data is quantification of information.

- **In the past, you would just go for a morning jog.**
- **Today, you know it was 7.6km long, you took 11,341 steps and burned 612 calories because of it.**

Definition and Characteristics of Big Data

*“Big data is **high-volume**, **high-velocity** and/or **high-variety** information assets that demand **cost-effective**, **innovative** forms of information processing that enable **enhanced insight, decision making, and process automation**.”*

– Gartner, Inc.

which was derived from:

*“While enterprises struggle to consolidate systems and collapse redundant databases to enable greater operational, analytical, and collaborative consistencies, changing economic conditions have made this job more difficult. E-commerce, in particular, has exploded data management challenges along three dimensions: **volumes, velocity and variety**. In 2001/02, IT organizations much compile a variety of approaches to have at their disposal for dealing each.”*

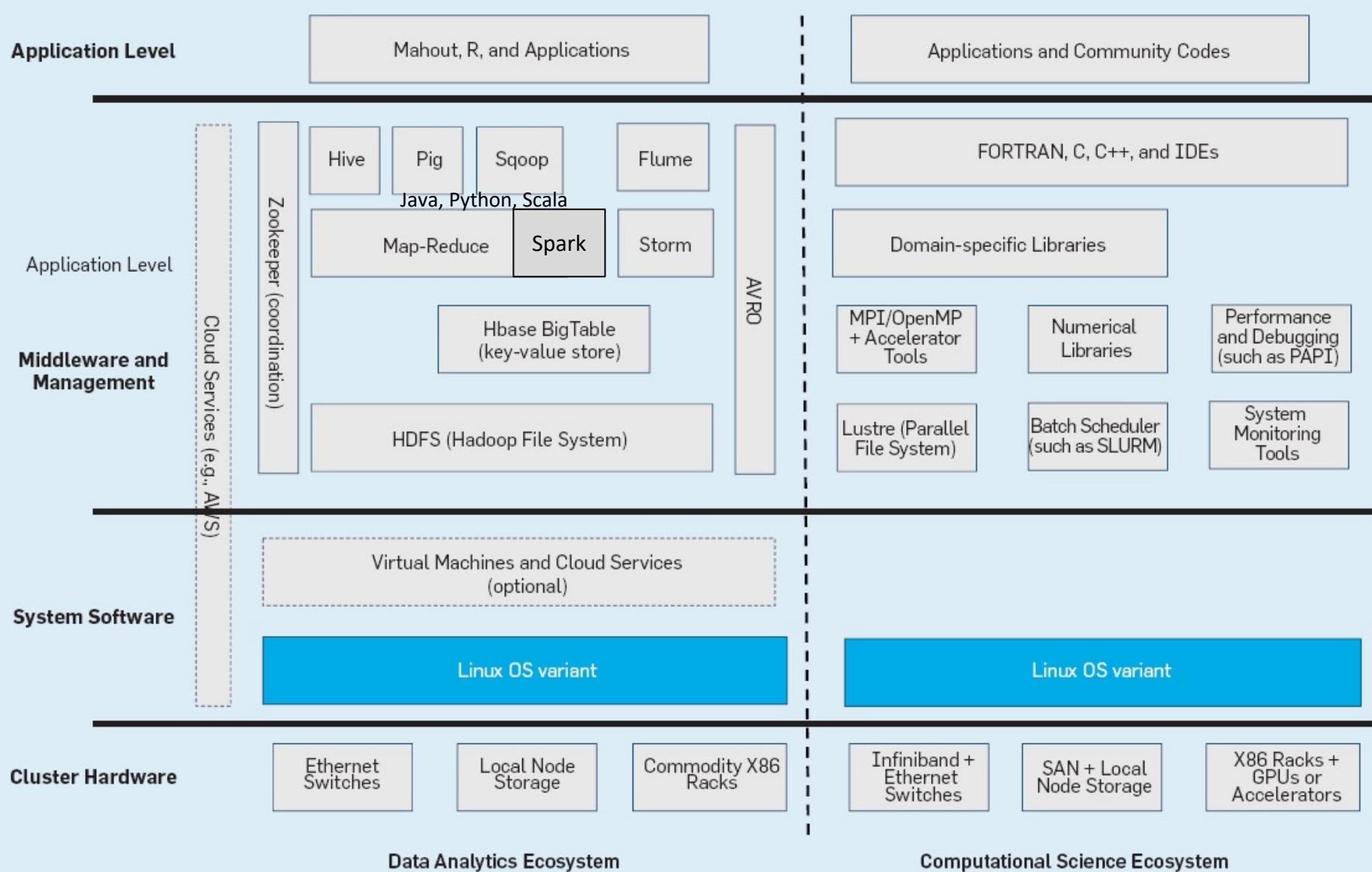
– Doug Laney

Comparison of Approaches in Adopting High-Performance Capabilities

Aspect	Typical Scenario	Big Data
Application development	Applications that take advantage of massive parallelism developed by specialized developers skilled in high-performance computing, performance optimization, and code tuning	A simplified application execution model encompassing a distributed file system, application programming model, distributed database, and program scheduling is packaged within Hadoop, an open source framework for reliable, scalable, distributed, and parallel computing
Platform	Uses high-cost massively parallel processing (MPP) computers, utilizing high-bandwidth networks, and massive I/O devices	Innovative methods of creating scalable and yet elastic virtualized platforms take advantage of clusters of commodity hardware components (either cycle harvesting from local resources or through cloud-based utility computing services) coupled with open source tools and technology
Data management	Limited to file-based or relational database management systems (RDBMS) using standard row-oriented data layouts	Alternate models for data management (often referred to as NoSQL or “Not Only SQL”) provide a variety of methods for managing information to best suit specific business process needs, such as in-memory data management (for rapid access), columnar layouts to speed query response, and graph databases (for social network analytics)
Resources	Requires large capital investment in purchasing high-end hardware to be installed and managed in-house	The ability to deploy systems like Hadoop on virtualized platforms allows small and medium businesses to utilize cloud-based environments that, from both a cost accounting and a practical perspective, are much friendlier to the bottom line

“Big Data Analytics”, David Loshin, 2013

Comparison of Data Analytics and Computing Ecosystems



Apache Hadoop



The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing.

The Apache Hadoop software library is **a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models**. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than relying on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

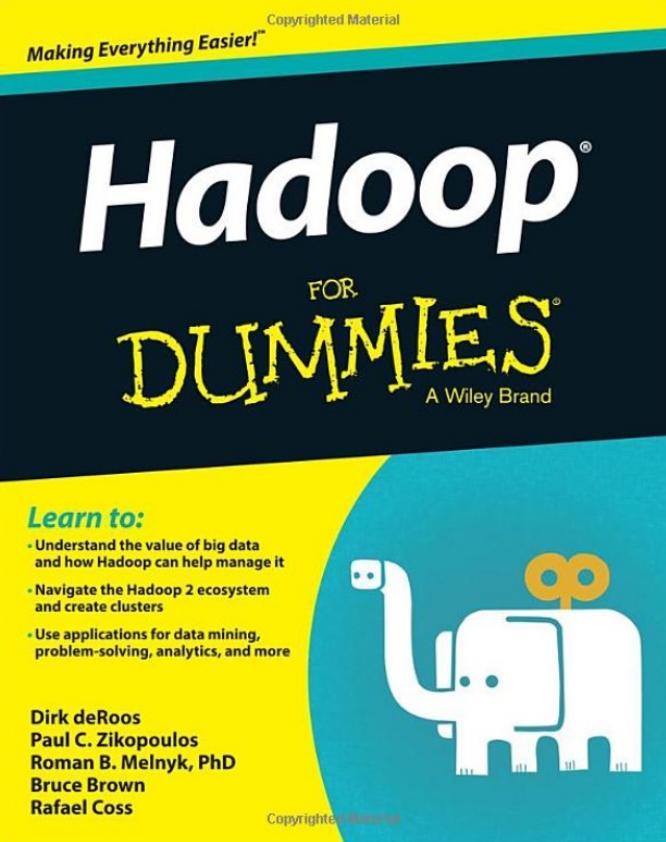
- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.
- **Hadoop YARN (starting from the 2nd generation)**: A framework for job scheduling and cluster resource management.

<http://hadoop.apache.org>

Hadoop-related Apache Projects: Hadoop Ecosystem

- **Ambari™**: A web-based tool for provisioning, managing, and monitoring Hadoop clusters. It also provides a dashboard for viewing cluster health and ability to view MapReduce, Pig and Hive applications visually.
- **Avro™**: A data serialization system.
- **Cassandra™**: A scalable multi-master database with no single points of failure.
- **Chukwa™**: A data collection system for managing large distributed systems.
- **HBase™**: A scalable, distributed database that supports structured data storage for large tables.
- **Hive™**: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- **Mahout™**: A scalable machine learning and data mining library.
- **Pig™**: A high-level data-flow language and execution framework for parallel computation.
- **Spark™**: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.
- **Tez™**: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases.
- **Zookeeper™**: A high-performance coordination service for distributed applications.

Reading Reference



Introduction 1

Part I: Getting Started with Hadoop 7

Chapter 1: Introducing Hadoop and Seeing What It's Good For.....	9
Chapter 2: Common Use Cases for Big Data in Hadoop.....	23
Chapter 3: Setting Up Your Hadoop Environment.....	41

Part II: How Hadoop Works 51

Chapter 4: Storing Data in Hadoop: The Hadoop Distributed File System.....	53
Chapter 5: Reading and Writing Data.....	69
Chapter 6: MapReduce Programming	83
Chapter 7: Frameworks for Processing Data in Hadoop: YARN and MapReduce.....	103
Chapter 8: Pig: Hadoop Programming Made Easier	117
Chapter 9: Statistical Analysis in Hadoop.....	129
Chapter 10: Developing and Scheduling Application Workflows with Oozie.....	139

Part III: Hadoop and Structured Data 155

Chapter 11: Hadoop and the Data Warehouse: Friends or Foes?	157
Chapter 12: Extremely Big Tables: Storing Data in HBase.....	179
Chapter 13: Applying Structure to Hadoop Data with Hive.....	227
Chapter 14: Integrating Hadoop with Relational Databases Using Sqoop.....	269
Chapter 15: The Holy Grail: Native SQL Access to Hadoop Data	303

Part IV: Administering and Configuring Hadoop 313

Chapter 16: Deploying Hadoop	315
Chapter 17: Administering Your Hadoop Cluster.....	335

Part V: The Part of Tens 359

Chapter 18: Ten Hadoop Resources Worthy of a Bookmark	361
Chapter 19: Ten Reasons to Adopt Hadoop	371

Big Data (Hadoop) Ecosystem

Big Data Applications/Domains

(Healthcare, insurance, finance, social networks,
transportation, sciences, etc.)

Big Data Analytics

(Methods: AI, machine learning, visualization, etc.
Modules: Pig, Hive, Mahout, etc.)

Big Data Computing

(MapReduce, Spark, Storm, Oozie, etc.)

Resource Management and Scheduling

(YARN, Kubernetes, Mesos)

Big Data Management

(NoSQL: RDBMS, Key-Value, Document, Graph, etc.
Systems: SQL, MongoDB, HBase, Cassandra, etc.)

Big Data Storage

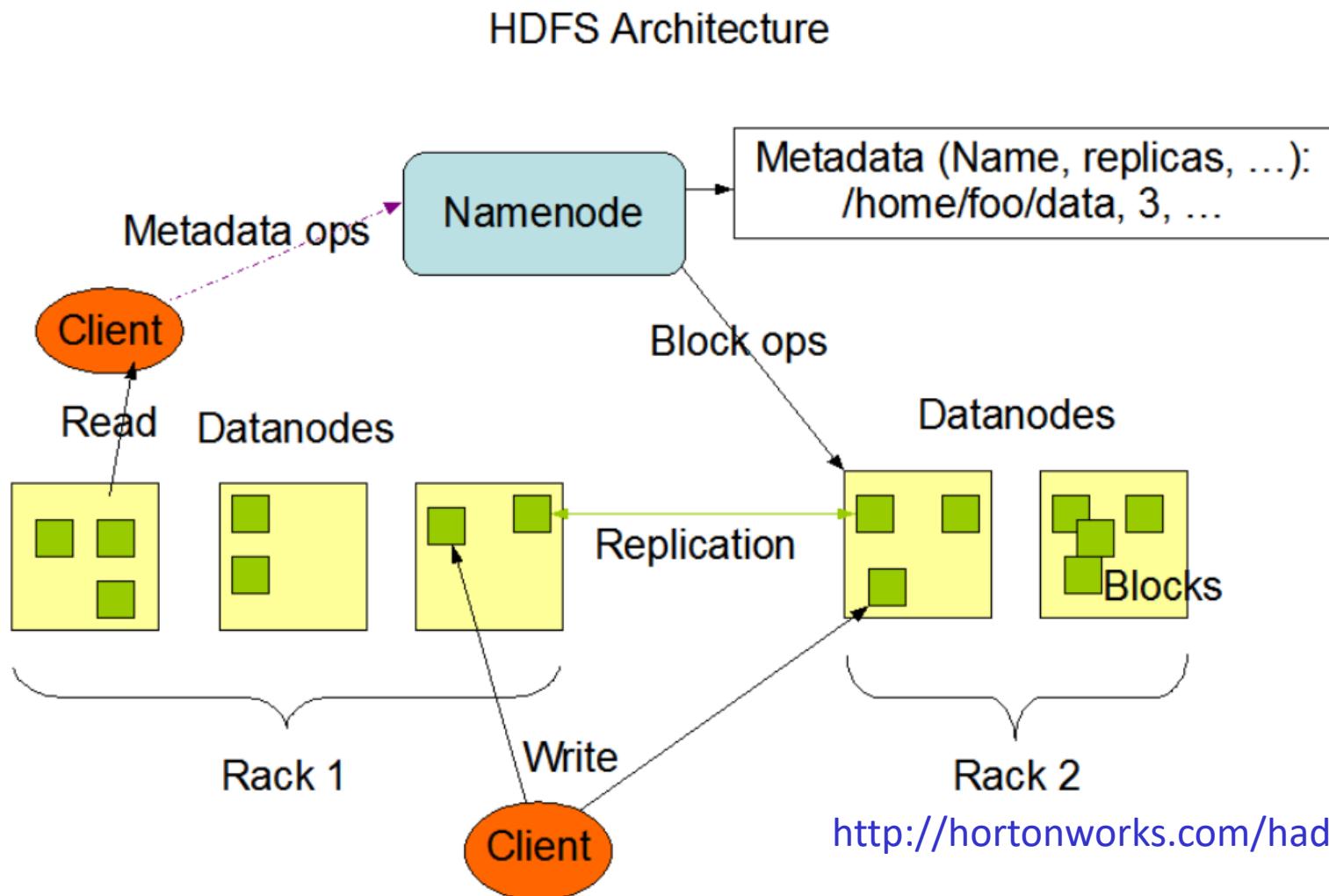
(HDFS)

Big Data Networking

(HPN, SDN, etc.)

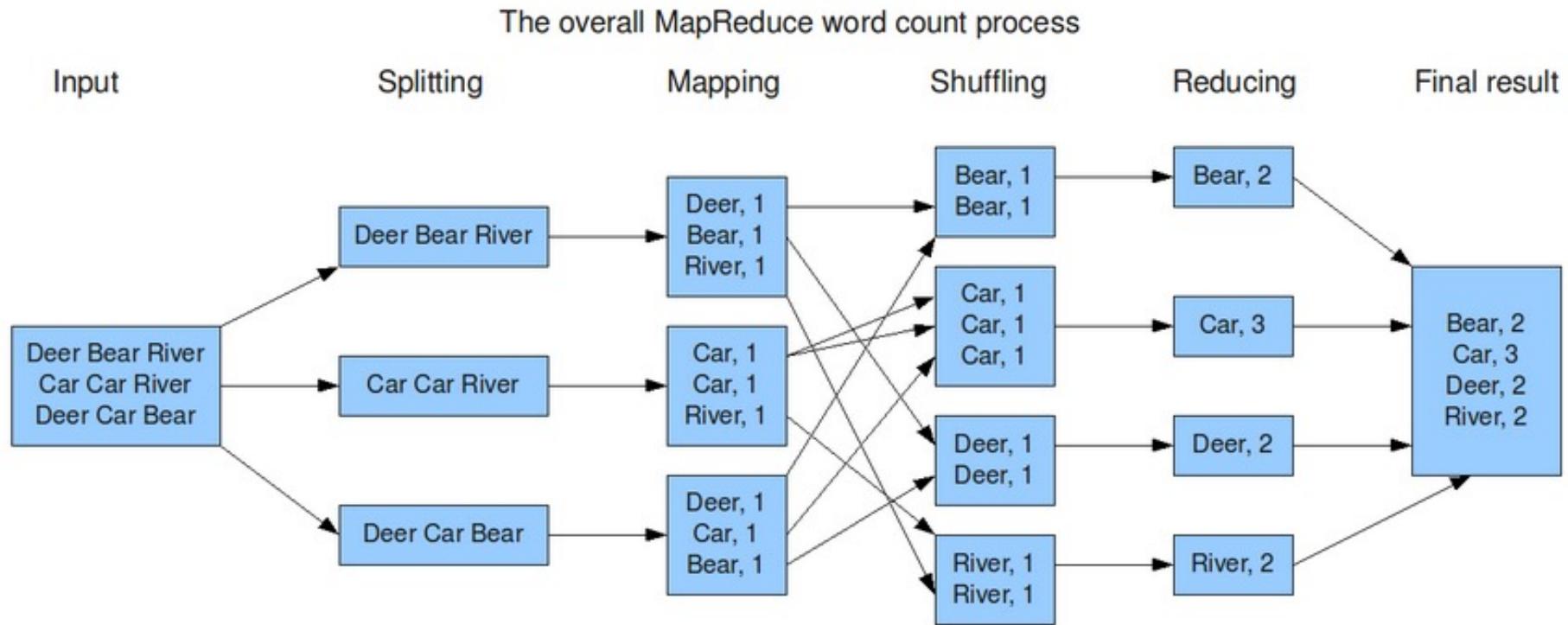
Hadoop Distributed File System (HDFS)

- HDFS is a java-based file system that provides the scalable, fault-tolerant, cost-efficient storage for big data
 - The file content is split into large blocks (typically 128 megabytes), each of which is independently replicated at multiple DataNodes
 - The NameNode maintains the namespace tree (in RAM) and the mapping of blocks to DataNodes



WordCounting: “Hello World” in MapReduce

Basic data structure: (key, value)



Set Up the Hadoop Environment

- Local (standalone) mode
- Pseudo-distributed mode
- Fully-distributed mode

Setting Up the Hadoop Environment – Pseudo-distributed mode

Configuration

Use the following:

conf/core-site.xml:

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

conf/hdfs-site.xml:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

conf/mapred-site.xml:

```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
</configuration>
```

Setup passphraseless ssh

Now check that you can ssh to the localhost without a passphrase:

```
$ ssh localhost
```

If you cannot ssh to localhost without a passphrase, execute the following commands:

```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

On the SSH server
authorized_keys:

used by the SSH server to store
the public keys of clients for client
authentication

On the SSH client
known_hosts:

used by the SSH client to store
the public keys of servers for
server authentication

Set Up the Hadoop Environment – Pseudo-distributed mode

Execution

Format a new distributed-filesystem:

```
$ bin/hadoop namenode -format
```

Start the hadoop daemons:

```
$ bin/start-all.sh
```

The hadoop daemon log output is written to the `$HADOOP_LOG_DIR` directory (defaults to `$HADOOP_PREFIX/logs`).

Browse the web interface for the NameNode and the JobTracker; by default they are available at:

- NameNode - <http://localhost:50070/>
- JobTracker - <http://localhost:50030/>

Copy the input files into the distributed filesystem:

```
$ bin/hadoop fs -put conf input
```

Run some of the examples provided:

```
$ bin/hadoop jar hadoop-*--examples.jar grep input output 'dfs[a-z.]+'
```

Examine the output files:

Copy the output files from the distributed filesystem to the local filesystem and examine them:

```
$ bin/hadoop fs -get output output  
$ cat output/*
```

or

View the output files on the distributed filesystem:

2:

```
$ bin/hadoop fs -cat output/*
```

Word Count Problem: Hands-on MapReduce Programming Guide -Configuration

Version: Hadoop 1.2.1

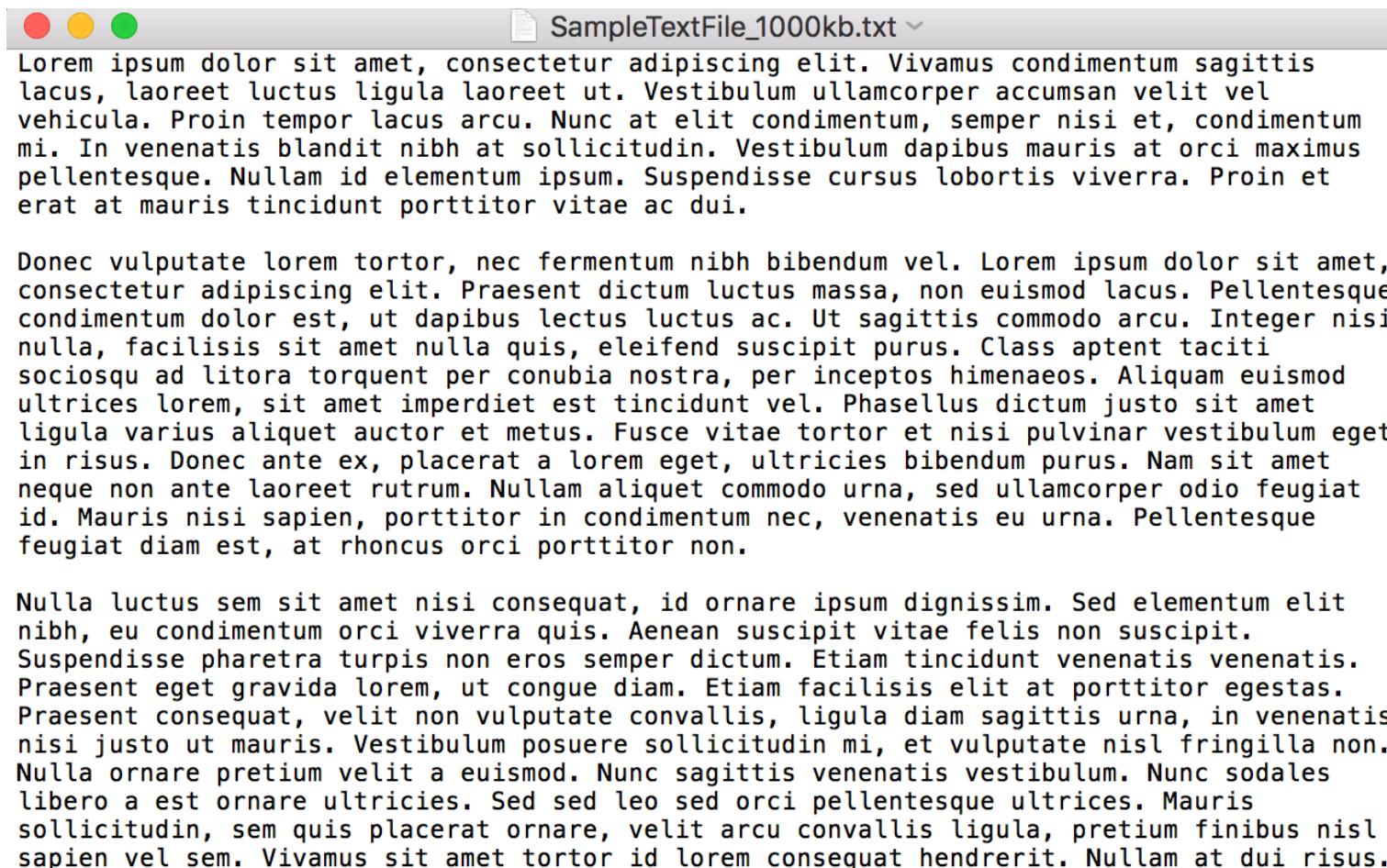
Mode: Pseudo-Distributed Mode

IDE: Eclipse

Word Count Problem

-Input

Locally stored file: SampleTextFile_1000kb.txt



The screenshot shows a Mac OS X window titled "SampleTextFile_1000kb.txt". The window contains three distinct blocks of placeholder text (Lorem ipsum) in a monospaced font. The first block is at the top, the second is in the middle, and the third is at the bottom. Each block consists of approximately 1000 lines of text.

```
SampleTextFile_1000kb.txt

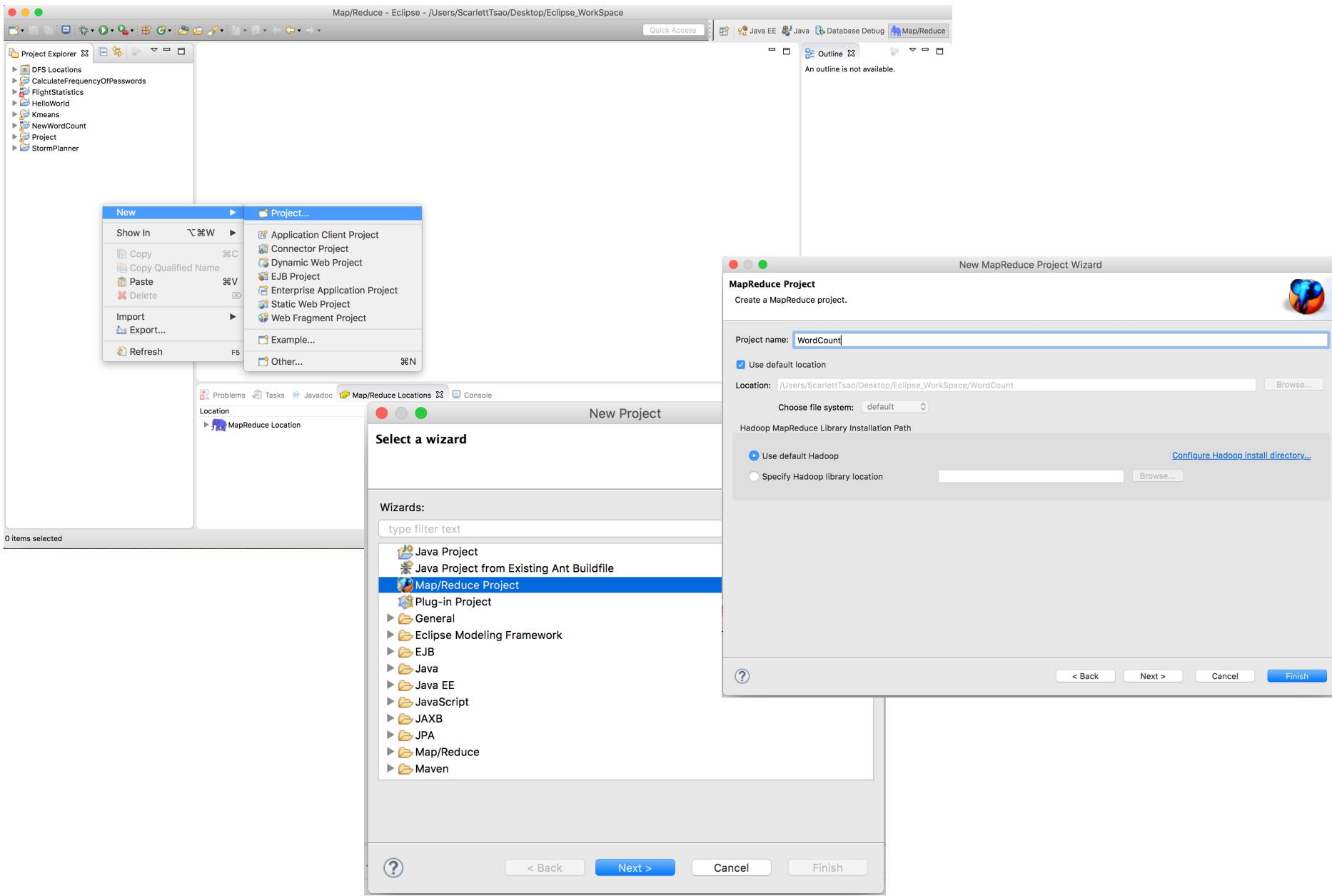
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus condimentum sagittis
lacus, laoreet luctus ligula laoreet ut. Vestibulum ullamcorper accumsan velit vel
vehicula. Proin tempor lacus arcu. Nunc at elit condimentum, semper nisi et, condimentum
mi. In venenatis blandit nibh at sollicitudin. Vestibulum dapibus mauris at orci maximus
pellentesque. Nullam id elementum ipsum. Suspendisse cursus lobortis viverra. Proin et
erat at mauris tincidunt porttitor vitae ac dui.

Donec vulputate lorem tortor, nec fermentum nibh bibendum vel. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Praesent dictum luctus massa, non euismod lacus. Pellentesque
condimentum dolor est, ut dapibus lectus luctus ac. Ut sagittis commodo arcu. Integer nisi
nulla, facilisis sit amet nulla quis, eleifend suscipit purus. Class aptent taciti
sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam euismod
ultrices lorem, sit amet imperdiet est tincidunt vel. Phasellus dictum justo sit amet
ligula varius aliquet auctor et metus. Fusce vitae tortor et nisi pulvinar vestibulum eget
in risus. Donec ante ex, placerat a lorem eget, ultricies bibendum purus. Nam sit amet
neque non ante laoreet rutrum. Nullam aliquet commodo urna, sed ullamcorper odio feugiat
id. Mauris nisi sapien, porttitor in condimentum nec, venenatis eu urna. Pellentesque
feugiat diam est, at rhoncus orci porttitor non.

Nulla luctus sem sit amet nisi consequat, id ornare ipsum dignissim. Sed elementum elit
nibh, eu condimentum orci viverra quis. Aenean suscipit vitae felis non suscipit.
Suspendisse pharetra turpis non eros semper dictum. Etiam tincidunt venenatis venenatis.
Praesent eget gravida lorem, ut congue diam. Etiam facilisis elit at porttitor egestas.
Praesent consequat, velit non vulputate convallis, ligula diam sagittis urna, in venenatis
nisi justo ut mauris. Vestibulum posuere sollicitudin mi, et vulputate nisl fringilla non.
Nulla ornare pretium velit a euismod. Nunc sagittis venenatis vestibulum. Nunc sodales
libero a est ornare ultricies. Sed sed leo sed orci pellentesque ultrices. Mauris
sollicitudin, sem quis placerat ornare, velit arcu convallis ligula, pretium finibus nisl
sapien vel sem. Vivamus sit amet tortor id lorem consequat hendrerit. Nullam at dui risus.
```

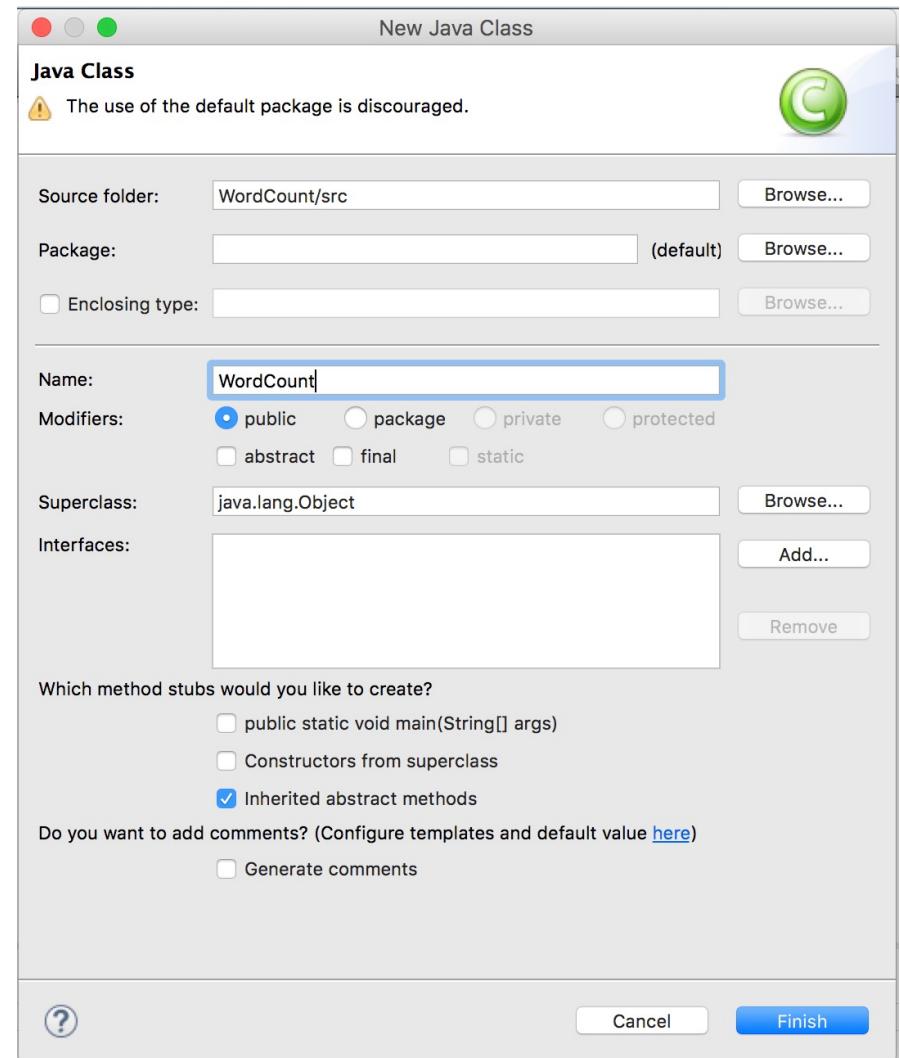
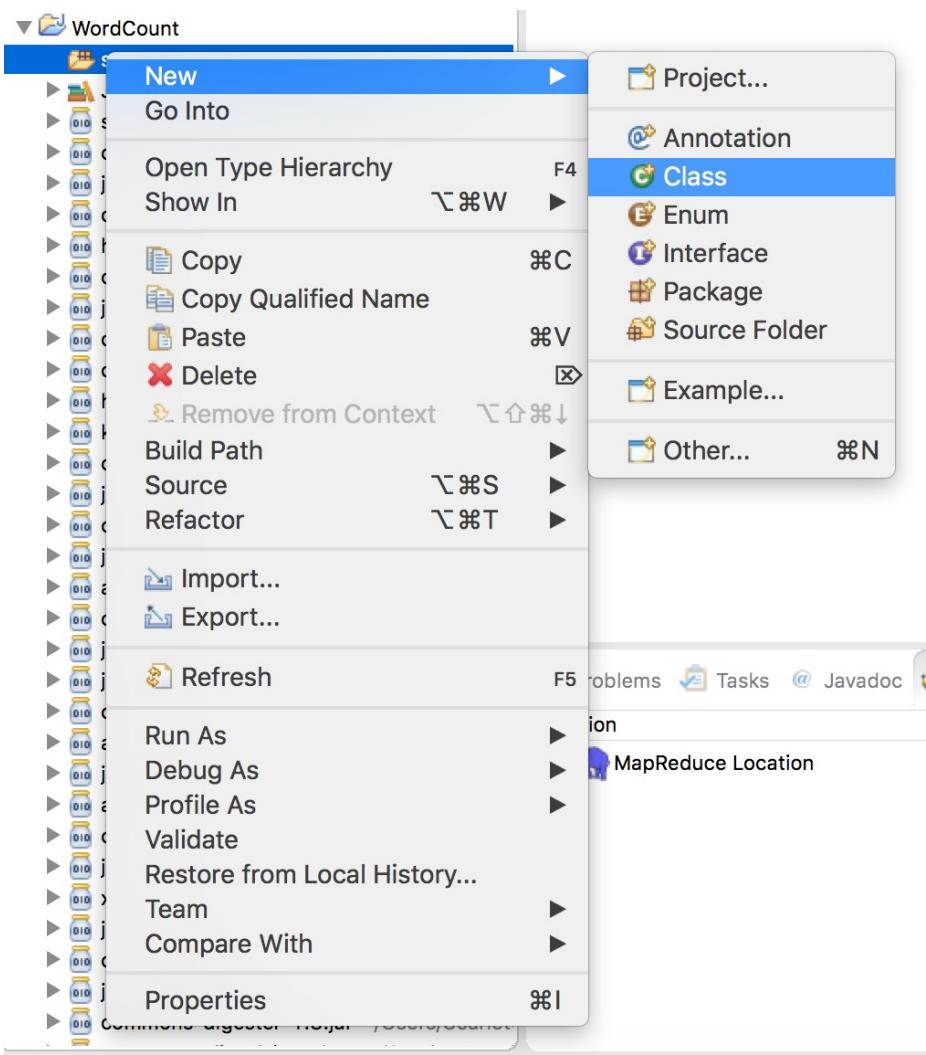
Word Count Problem

-Create a MapReduce Project



Word Count Problem

-Create a class



Word Count Problem -MapReduce Program

Mapper Function:

```
*WordCount.java ✎
1 import java.io.IOException;
2 import java.util.StringTokenizer;
3
4 import org.apache.hadoop.conf.Configuration;
5 import org.apache.hadoop.fs.Path;
6 import org.apache.hadoop.io.IntWritable;
7 import org.apache.hadoop.io.Text;
8 import org.apache.hadoop.mapreduce.Job;
9 import org.apache.hadoop.mapreduce.Mapper;
10 import org.apache.hadoop.mapreduce.Reducer;
11 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
12 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
13 import org.apache.hadoop.util.GenericOptionsParser;
14
15 public class WordCount {
16
17     public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>{
18         //mapper function
19         private final static IntWritable one = new IntWritable(1);
20         private Text word = new Text();
21
22         public void map(Object key, Text value, Context context) throws IOException, InterruptedException {
23             StringTokenizer itr = new StringTokenizer(value.toString()); //convert text into string token iterator
24             while (itr.hasMoreTokens()) { //for each word generate the pair <word, one> as output context
25                 word.set(itr.nextToken());
26                 context.write(word, one);
27             }
28         }
29     }
30 }
```

Word Count Problem -MapReduce Program

Reducer Function:

```
*WordCount.java ✎
```

```
30
31  public static class IntSumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
32      //reducer function
33      private IntWritable result = new IntWritable();
34
35  public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {
36      int sum = 0;
37      for (IntWritable val : values) { //for each word, collect the "ones" and output the pair <keyword, result>
38          sum += val.get();
39      }
40      result.set(sum);
41      context.write(key, result);
42  }
43 }
44 }
```

Word Count Problem -MapReduce Program

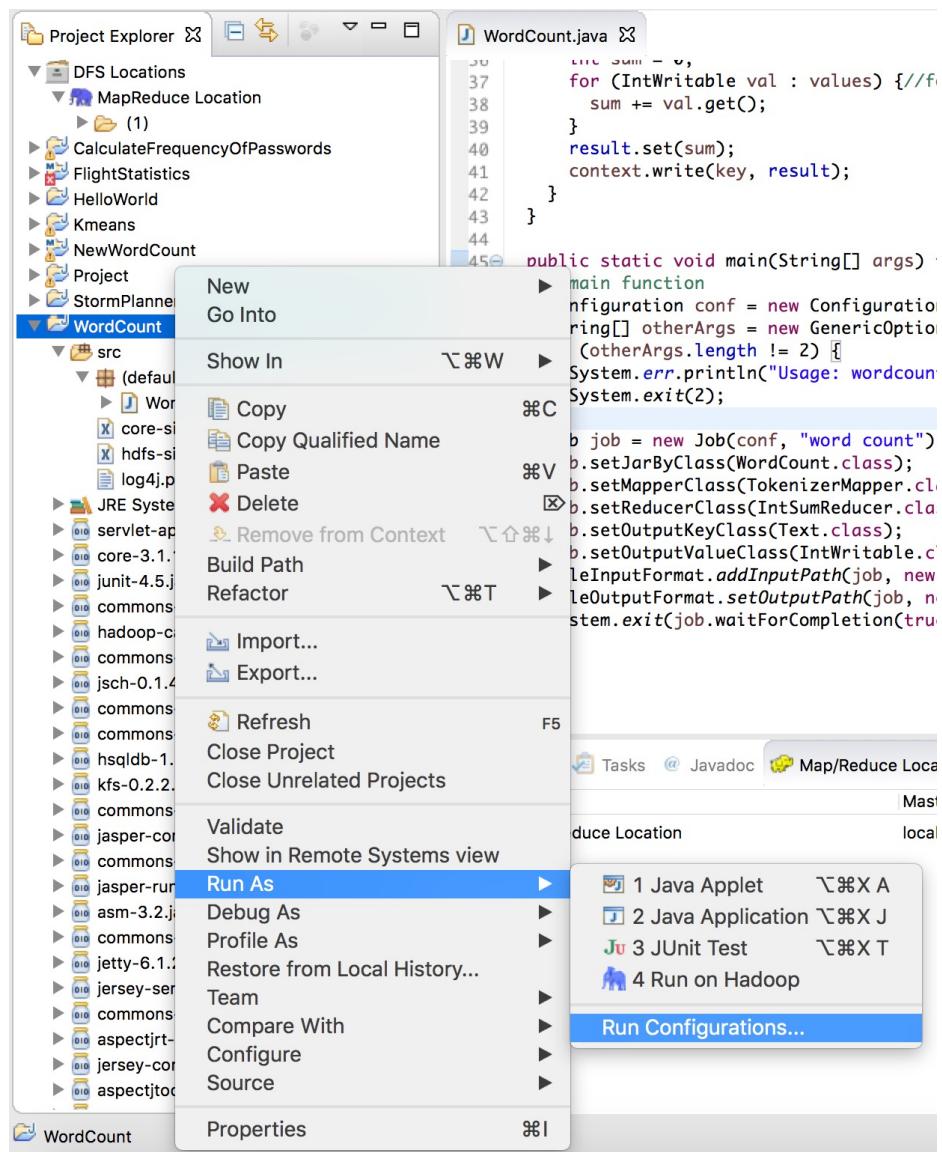
Main Function:

The screenshot shows a Java code editor window with the file *WordCount.java open. The code is a MapReduce program for word counting. The main function handles command-line arguments, initializes a Job object, and sets various configurations like input and output paths. A specific line of code, `job.setReducerClass(IntSumReducer.class);`, is highlighted with a light blue background.

```
45  public static void main(String[] args) throws Exception {
46      //main function
47      Configuration conf = new Configuration();
48      String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
49      if (otherArgs.length != 2) {
50          System.err.println("Usage: wordcount <in> <out>");
51          System.exit(2);
52      }
53      Job job = new Job(conf, "word count");
54      job.setJarByClass(WordCount.class);
55      job.setMapperClass(TokenizerMapper.class);
56      job.setReducerClass(IntSumReducer.class); // Line 56 is highlighted
57      job.setOutputKeyClass(Text.class);
58      job.setOutputValueClass(IntWritable.class);
59      FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
60      FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
61      System.exit(job.waitForCompletion(true) ? 0 : 1);
62  }
63 }
```

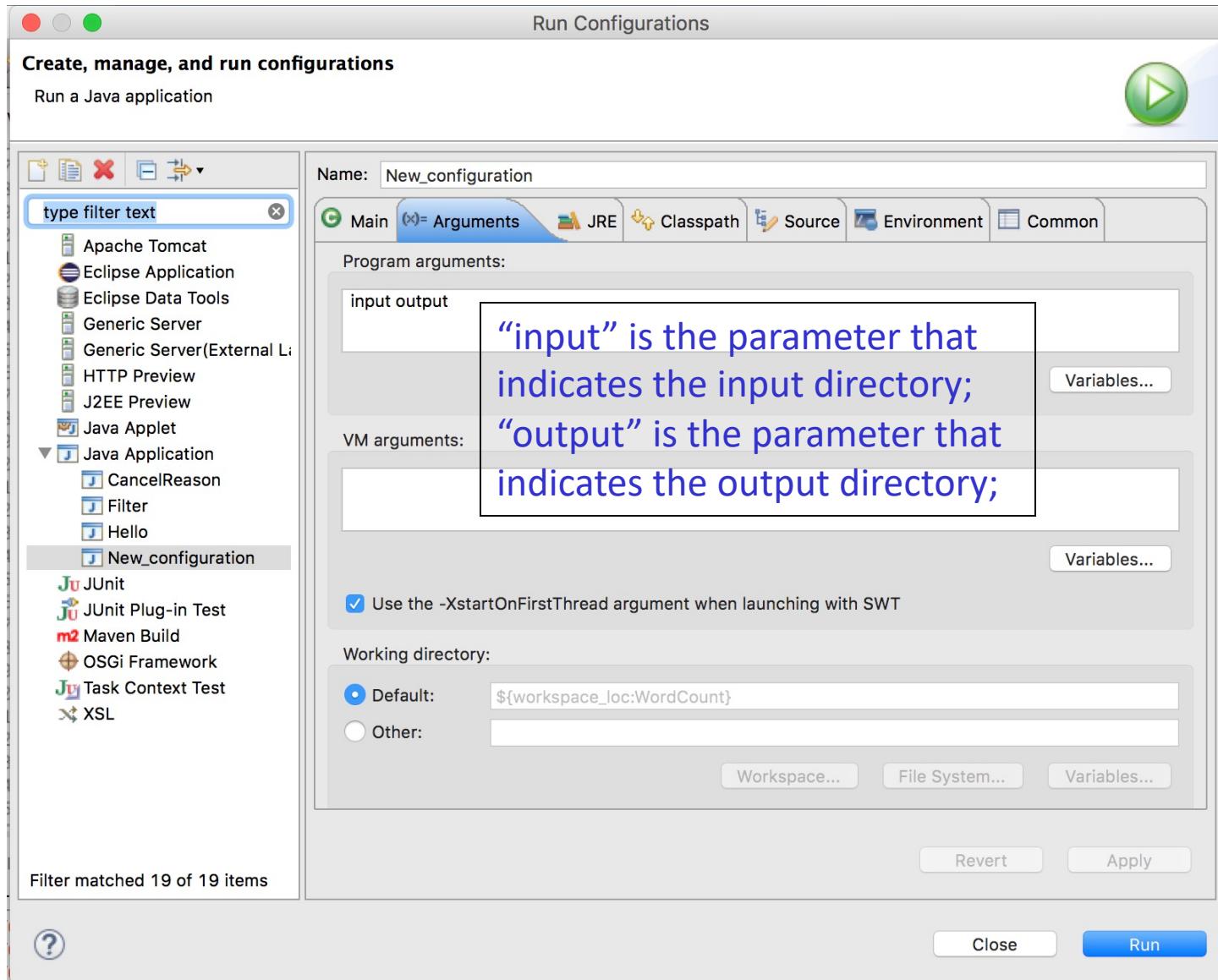
Word Count Problem

-Execute MapReduce Program on Eclipse



Word Count Problem

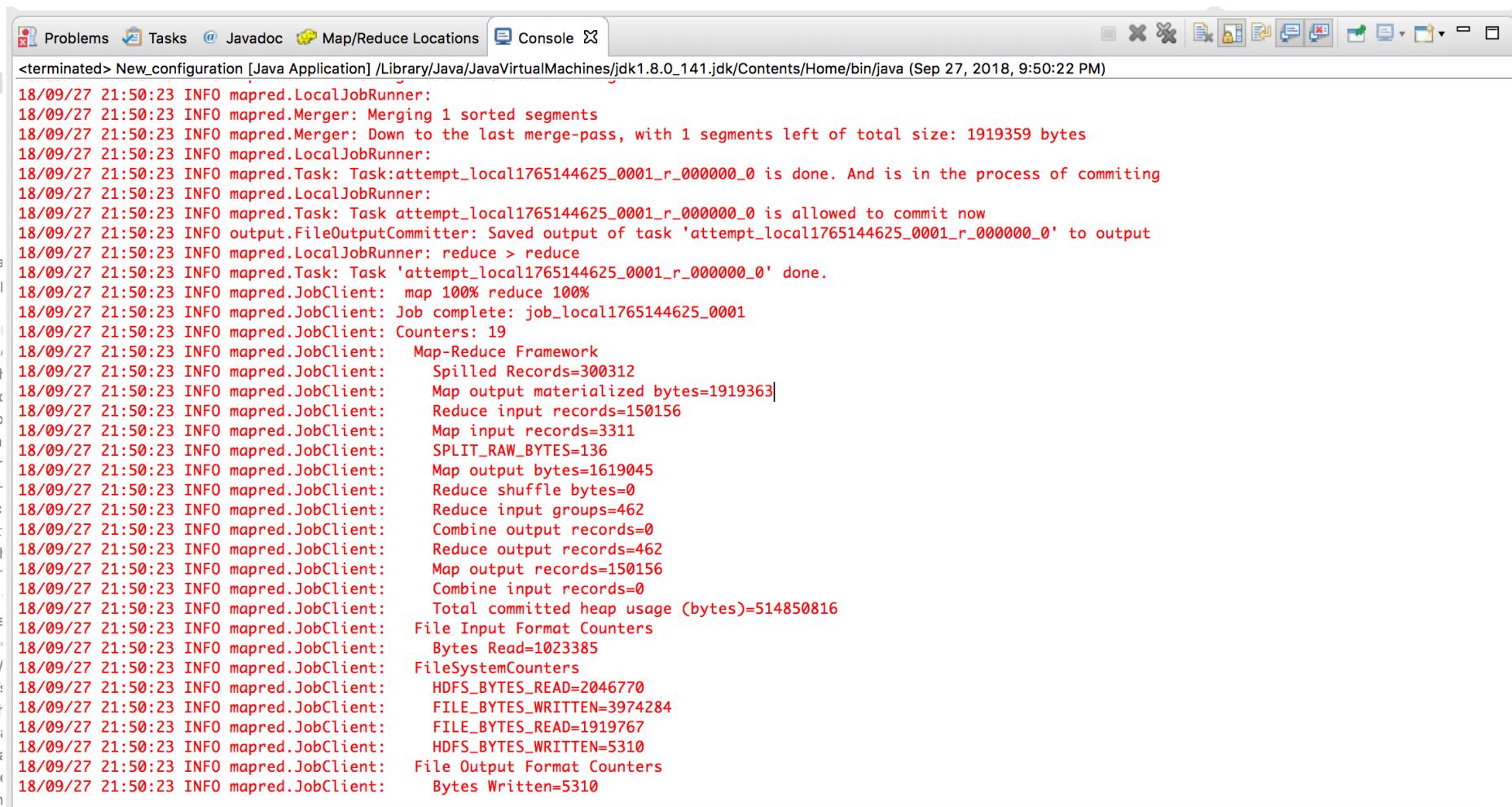
-Execute MapReduce Program on Eclipse



Word Count Problem

-Execute MapReduce Program on Eclipse

When the program is successfully executed:



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The console window displays log output from a Java application named 'New_configuration'. The log entries are in red text, indicating informational messages from the MapReduce framework. The log details the execution of a job, including merging segments, committing tasks, and summarizing job statistics like map/reduce counts, spilled records, and total heap usage.

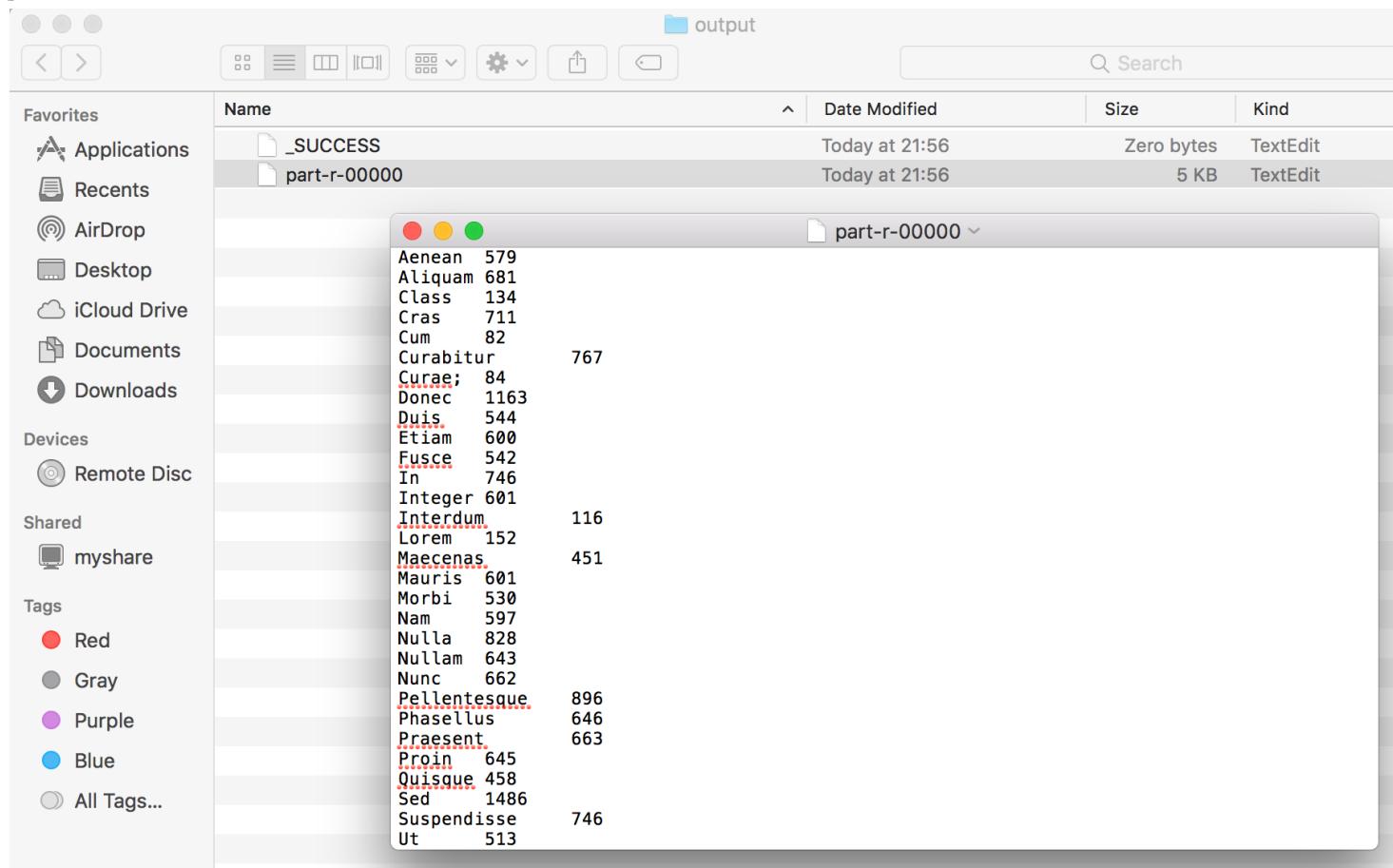
```
<terminated> New_configuration [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_141.jdk/Contents/Home/bin/java (Sep 27, 2018, 9:50:22 PM)
18/09/27 21:50:23 INFO mapred.LocalJobRunner:
18/09/27 21:50:23 INFO mapred.Merger: Merging 1 sorted segments
18/09/27 21:50:23 INFO mapred.Merger: Down to the last merge-pass, with 1 segments left of total size: 1919359 bytes
18/09/27 21:50:23 INFO mapred.LocalJobRunner:
18/09/27 21:50:23 INFO mapred.Task: Task:attempt_local1765144625_0001_r_000000_0 is done. And is in the process of committing
18/09/27 21:50:23 INFO mapred.LocalJobRunner:
18/09/27 21:50:23 INFO mapred.Task: Task attempt_local1765144625_0001_r_000000_0 is allowed to commit now
18/09/27 21:50:23 INFO output.FileOutputCommitter: Saved output of task 'attempt_local1765144625_0001_r_000000_0' to output
18/09/27 21:50:23 INFO mapred.LocalJobRunner: reduce > reduce
18/09/27 21:50:23 INFO mapred.Task: Task 'attempt_local1765144625_0001_r_000000_0' done.
18/09/27 21:50:23 INFO mapred.JobClient: map 100% reduce 100%
18/09/27 21:50:23 INFO mapred.JobClient: Job complete: job_local1765144625_0001
18/09/27 21:50:23 INFO mapred.JobClient: Counters: 19
18/09/27 21:50:23 INFO mapred.JobClient: Map-Reduce Framework
18/09/27 21:50:23 INFO mapred.JobClient:   Spilled Records=300312
18/09/27 21:50:23 INFO mapred.JobClient:   Map output materialized bytes=1919363|
18/09/27 21:50:23 INFO mapred.JobClient:   Reduce input records=150156
18/09/27 21:50:23 INFO mapred.JobClient:   Map input records=3311
18/09/27 21:50:23 INFO mapred.JobClient:   SPLIT_RAW_BYTES=136
18/09/27 21:50:23 INFO mapred.JobClient:   Map output bytes=1619045
18/09/27 21:50:23 INFO mapred.JobClient:   Reduce shuffle bytes=0
18/09/27 21:50:23 INFO mapred.JobClient:   Reduce input groups=462
18/09/27 21:50:23 INFO mapred.JobClient:   Combine output records=0
18/09/27 21:50:23 INFO mapred.JobClient:   Reduce output records=462
18/09/27 21:50:23 INFO mapred.JobClient:   Map output records=150156
18/09/27 21:50:23 INFO mapred.JobClient:   Combine input records=0
18/09/27 21:50:23 INFO mapred.JobClient:   Total committed heap usage (bytes)=514850816
18/09/27 21:50:23 INFO mapred.JobClient: File Input Format Counters
18/09/27 21:50:23 INFO mapred.JobClient:   Bytes Read=1023385
18/09/27 21:50:23 INFO mapred.JobClient: FileSystemCounters
18/09/27 21:50:23 INFO mapred.JobClient:   HDFS_BYTES_READ=2046770
18/09/27 21:50:23 INFO mapred.JobClient:   FILE_BYTES_WRITTEN=3974284
18/09/27 21:50:23 INFO mapred.JobClient:   FILE_BYTES_READ=1919767
18/09/27 21:50:23 INFO mapred.JobClient:   HDFS_BYTES_WRITTEN=5310
18/09/27 21:50:23 INFO mapred.JobClient: File Output Format Counters
18/09/27 21:50:23 INFO mapred.JobClient:   Bytes Written=5310
```

Word Count Problem

-Check the output

```
[Scarletts-MBP:hadoop-1.2.1 ScarlettTsao$ hadoop fs -get output ~/Desktop/  
Warning: $HADOOP_HOME is deprecated.
```

```
Scarletts-MBP:hadoop-1.2.1 ScarlettTsao$
```



Word Count Problem

-Configuration

Version: Hadoop 1.2.1

Mode: Fully-Distributed Mode

Cloud: Amazon Web Service

Word Count Problem -Configuration

Three homogenous VM instances: one master node, and two slave nodes.

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (which is selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Images, AMIs, and Bundle Tasks. Below that are sections for Elastic Block Store Volumes and Snapshots, and Network & Security Security Groups, Elastic IPs, Placement Groups, and Key Pairs. At the bottom, there are Feedback and English (US) buttons, along with links to Privacy Policy and Terms of Use.

The main content area displays a list of three instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
master	i-0bd7925a4386838...	t2.micro	us-west-1c	running	2/2 checks ...	None
slave1	i-0b3204e39bb520a...	t2.micro	us-west-1c	running	2/2 checks ...	None
slave2	i-09b612c0983d4c672	t2.micro	us-west-1c	running	2/2 checks ...	None

Below the instance list, there's a message "Select an instance above" and three small icons.

Instance Type

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Instance Details

Storage

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snap-0c44a2efc327e7ee6	8	gp2	100 / 3000	N/A	Yes	Not Encrypted

Word Count Problem

-Input

Create the input directory in HDFS and upload the input data from local to this directory. The input size is 108MB.

```
[...].ssh — hadoop@ip-10-0-0-67:~ — ssh -i hadoop.pem ec2-user@master — 127x53
...67:~ — ssh -i hadoop.pem ec2-user@master      ...ent — ssh -i hadoop.pem ec2-user@slave1      ...t — ssh -i hadoop.pem ec2-user@slave2 +[+]
[hadoop@ip-10-0-0-67 ~]$ hadoop fs -mkdir input
Warning: $HADOOP_HOME is deprecated.

[hadoop@ip-10-0-0-67 ~]$ ls
hadoop-1.2.1  jdk1.7.0_79  output  test  WordCount.jar
[hadoop@ip-10-0-0-67 ~]$ hadoop fs -put test input
Warning: $HADOOP_HOME is deprecated.

[hadoop@ip-10-0-0-67 ~]$ hadoop fs -ls input
Warning: $HADOOP_HOME is deprecated.

Found 1 items
-rw-r--r--  1 hadoop supergroup  108385536 2018-10-01 01:40 /user/hadoop/input/test
[hadoop@ip-10-0-0-67 ~]$
```

Word Count Problem

-Export JAR file with Eclipse

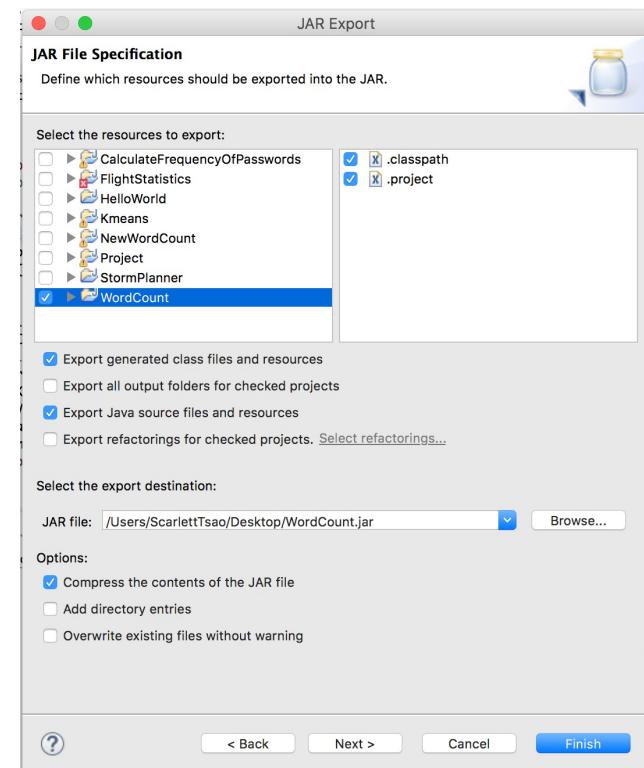
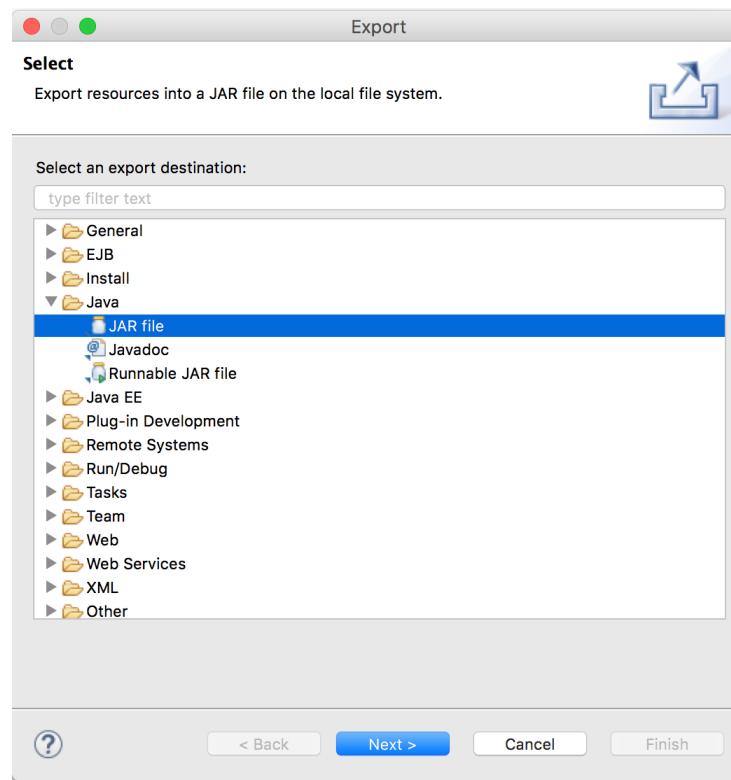
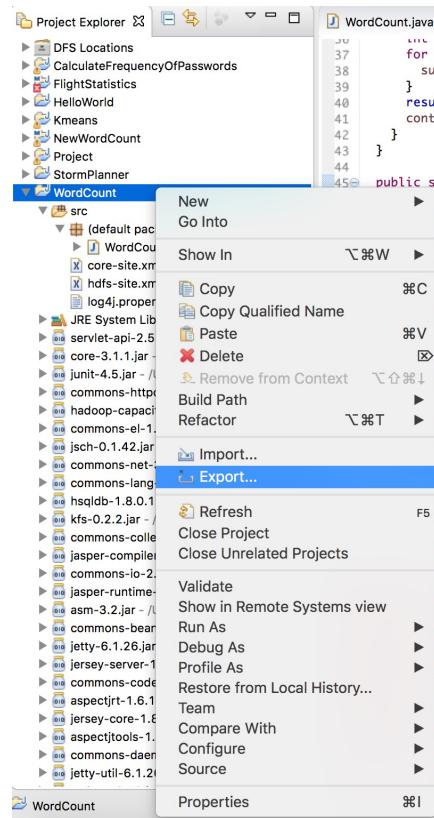
We use Eclipse to test the program locally (Stand-alone mode or Pseudo-Distributed mode).

If we want to run a MapReduce program in a Fully-Distributed Mode on a Hadoop cluster, for example, in a public cloud environment, we can upload the JAR file to the master node of the cluster and execute the program by using the following command:

```
$ bin/hadoop jar WordCount.jar WordCount /user/user_name/wordcount/input  
/user/user_name/wordcount/output
```

Word Count Problem

-Export JAR file with Eclipse



Word Count Problem

-Execution

Execute JAR on the cluster.

```
● ○ ● .ssh — hadoop@ip-10-0-0-67:~ — ssh -i hadoop.pem ec2-user@master — 127x56
[hadoop@ip-10-0-0-67 ~]$ hadoop jar WordCount.jar WordCount input output
Warning: $HADOOP_HOME is deprecated.

18/10/01 00:53:13 INFO input.FileInputFormat: Total input paths to process : 1
18/10/01 00:53:13 INFO util.NativeCodeLoader: Loaded the native-hadoop library
18/10/01 00:53:13 WARN snappy.LoadSnappy: Snappy native library not loaded
18/10/01 00:53:14 INFO mapred.JobClient: Running job: job_201810010050_0001
18/10/01 00:53:15 INFO mapred.JobClient: map 0% reduce 0%
18/10/01 00:53:27 INFO mapred.JobClient: map 18% reduce 0%
18/10/01 00:53:28 INFO mapred.JobClient: map 50% reduce 0%
18/10/01 00:53:30 INFO mapred.JobClient: map 61% reduce 0%
18/10/01 00:53:31 INFO mapred.JobClient: map 80% reduce 0%
18/10/01 00:53:33 INFO mapred.JobClient: map 92% reduce 0%
18/10/01 00:53:36 INFO mapred.JobClient: map 100% reduce 0%
18/10/01 00:53:42 INFO mapred.JobClient: map 100% reduce 16%
18/10/01 00:53:45 INFO mapred.JobClient: map 100% reduce 70%
18/10/01 00:53:48 INFO mapred.JobClient: map 100% reduce 82%
18/10/01 00:53:51 INFO mapred.JobClient: map 100% reduce 93%
18/10/01 00:53:54 INFO mapred.JobClient: map 100% reduce 100%
18/10/01 00:53:55 INFO mapred.JobClient: Job complete: job_201810010050_0001
18/10/01 00:53:55 INFO mapred.JobClient: Counters: 30
18/10/01 00:53:55 INFO mapred.JobClient: Job Counters
18/10/01 00:53:55 INFO mapred.JobClient: Launched reduce tasks=1
18/10/01 00:53:55 INFO mapred.JobClient: SLOTS_MILLIS_MAPS=38850
18/10/01 00:53:55 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
18/10/01 00:53:55 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
18/10/01 00:53:55 INFO mapred.JobClient: Rack-local map tasks=1
18/10/01 00:53:55 INFO mapred.JobClient: Launched map tasks=2
18/10/01 00:53:55 INFO mapred.JobClient: Data-local map tasks=1
18/10/01 00:53:55 INFO mapred.JobClient: SLOTS_MILLIS_REDUCES=19652
18/10/01 00:53:55 INFO mapred.JobClient: File Output Format Counters
18/10/01 00:53:55 INFO mapred.JobClient: Bytes Written=135481920
18/10/01 00:53:55 INFO mapred.JobClient: FileSystemCounters
18/10/01 00:53:55 INFO mapred.JobClient: FILE_BYTES_READ=513974282
18/10/01 00:53:55 INFO mapred.JobClient: HDFS_BYTES_READ=108389844
18/10/01 00:53:55 INFO mapred.JobClient: FILE_BYTES_WRITTEN=703824550
18/10/01 00:53:55 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=135481920
18/10/01 00:53:55 INFO mapred.JobClient: File Input Format Counters
18/10/01 00:53:55 INFO mapred.JobClient: Bytes Read=108389632
18/10/01 00:53:55 INFO mapred.JobClient: Map-Reduce Framework
18/10/01 00:53:55 INFO mapred.JobClient: Map output materialized bytes=189674700
18/10/01 00:53:55 INFO mapred.JobClient: Map input records=13548192
18/10/01 00:53:55 INFO mapred.JobClient: Reduce shuffle bytes=189674700
18/10/01 00:53:55 INFO mapred.JobClient: Spilled Records=50260615
18/10/01 00:53:55 INFO mapred.JobClient: Map output bytes=162578304
18/10/01 00:53:55 INFO mapred.JobClient: Total committed heap usage (bytes)=336011264
18/10/01 00:53:55 INFO mapred.JobClient: CPU time spent (ms)=39640
18/10/01 00:53:55 INFO mapred.JobClient: Combine input records=0
18/10/01 00:53:55 INFO mapred.JobClient: SPLIT_RAW_BYTES=212
18/10/01 00:53:55 INFO mapred.JobClient: Reduce input records=13548192
18/10/01 00:53:55 INFO mapred.JobClient: Reduce input groups=13548192
18/10/01 00:53:55 INFO mapred.JobClient: Combine output records=0
18/10/01 00:53:55 INFO mapred.JobClient: Physical memory (bytes) snapshot=509358080
18/10/01 00:53:55 INFO mapred.JobClient: Reduce output records=13548192
18/10/01 00:53:55 INFO mapred.JobClient: Virtual memory (bytes) snapshot=2244427776
18/10/01 00:53:55 INFO mapred.JobClient: Map output records=13548192
```

Word Count Problem

-Output

Download the output folder from HDFS to the master node.

Finds out the size of the output, which is 130MB.

```
...67:~ — ssh -i hadoop.pem ec2-user@master — 127x53
...67:~ — ssh -i hadoop.pem ec2-user@master ...ent — ssh -i hadoop.pem ec2-user@slave1 ...t — ssh -i hadoop.pem ec2-user@slave2 +
```

```
[hadoop@ip-10-0-0-67 ~]$ hadoop fs -ls output
Warning: $HADOOP_HOME is deprecated.

Found 3 items
-rw-r--r-- 1 hadoop supergroup 0 2018-10-01 01:20 /user/hadoop/output/_SUCCESS
drwxr-xr-x - hadoop supergroup 0 2018-10-01 01:19 /user/hadoop/output/_logs
-rw-r--r-- 1 hadoop supergroup 135481920 2018-10-01 01:20 /user/hadoop/output/part-r-00000
[hadoop@ip-10-0-0-67 ~]$ hadoop fs -get output .
Warning: $HADOOP_HOME is deprecated.

[hadoop@ip-10-0-0-67 ~]$ du -m output
1      output/_logs/history
1      output/_logs
130    output
[hadoop@ip-10-0-0-67 ~]$ du -m output/*
1      output/_logs/history
1      output/_logs
130    output/part-r-00000
0      output/_SUCCESS
[hadoop@ip-10-0-0-67 ~]$
```

Word Count Problem

-Output Location

Note that by default the data block size is 64MB in Hadoop 1.2.1, and 128MB in Hadoop 2.

Check slave node 1: there is only one block stored on this node.

By checking the first 10 lines of the data block's contents, we see that the file stored on slave node 1 contains the mapping keys.

```
● ○ ● .ssh — hadoop@ip-10-0-0-178:~/hadoop-1.2.1/data/current — ssh -i hadoop.pem ec2-user@slave1 — 127x54
...put — ssh -i hadoop.pem ec2-user@master ...ent — ssh -i hadoop.pem ec2-user@slave1 ...t — ssh -i hadoop.pem ec2-user@slave2 +
```

```
[[hadoop@ip-10-0-0-178 current]$ du -m *
1      blk_87520988563512578
1      blk_87520988563512578_1001.meta
64     blk_-956885929074624978
1      blk_-956885929074624978_1002.meta
1      dncp_block_verification.log.curr
1      VERSION

[[hadoop@ip-10-0-0-178 current]$ head -10 blk_-956885929074624978
aaaaaaaa
aaaaaaab
aaaaaac
aaaaaad
aaaaaae
aaaaaaaf
aaaaaaag
aaaaaaah
aaaaaaai
aaaaaaaj

[hadoop@ip-10-0-0-178 current]$
```

Word Count Problem -Output Location

Check slave node 2: there are two data blocks stored on this node.

By checking the first 10 lines of these two data blocks' contents, we see that the data blocks stored on slave node 2 contain the output.

```
● ○ ● .ssh — hadoop@ip-10-0-0-115:~/hadoop-1.2.1/data/current — ssh -i hadoop.pem ec2-user@slave2 — 127x52
...67:~ — ssh -i hadoop.pem ec2-user@master      ...ent — ssh -i hadoop.pem ec2-user@slave1      ...t — ssh -i hadoop.pem ec2-user@slave2      +
[[hadoop@ip-10-0-0-115 current]$ du -m *
40    blk_-1812166875710558882
1    blk_-1812166875710558882_1002.meta
64    blk_6112797773231470484
1    blk_6112797773231470484_1011.meta
2    blk_6516737765071159401
1    blk_6516737765071159401_1011.meta
64    blk_8272876575694637448
1    blk_8272876575694637448_1011.meta
1    blk_8745688025233117434
1    blk_8745688025233117434_1010.meta
1    blk_9064072999373545110
1    blk_9064072999373545110_1012.meta
1    dncp_block_verification.log.curr
1    VERSION
[[hadoop@ip-10-0-0-115 current]$ head -10 blk_6112797773231470484
aaaaaaaa 1
aaaaaaab 1
aaaaaaac 1
aaaaaaad 1
aaaaaaae 1
aaaaaaaf 1
aaaaaaag 1
aaaaaaah 1
aaaaaaai 1
aaaaaaaj 1
[[hadoop@ip-10-0-0-115 current]$ head -10 blk_8272876575694637448
vja 1
aaorvjb 1
aaorvjc 1
aaorvjd 1
aaorvje 1
aaorvjf 1
aaorvjq 1
aaorvjh 1
aaorvji 1
aaorvjj 1
[hadoop@ip-10-0-0-115 current]$
```

Word Count Problem -Configuration

Version: Hadoop 2.6.0

Mode: Fully-Distributed Mode

Cloud: Amazon Web Service

Word Count Problem -Configuration

Three homogenous Virtual Machine instances:

One master node

Two slave nodes

The screenshot shows the AWS EC2 Dashboard with the following details:

- Instances:** 15 instances listed in a table.
- Master Node:** master (running, IP: 34.218.10.95)
- Slave Nodes:** slave1, slave2 (running, IP: 52.24.10.152), z1, z2, s7, r1, m1, m2, s1, s2, m3 (all stopped)
- Region:** Oregon (Oregon)
- Filter:** Filter by tags and attributes or search by keyword.

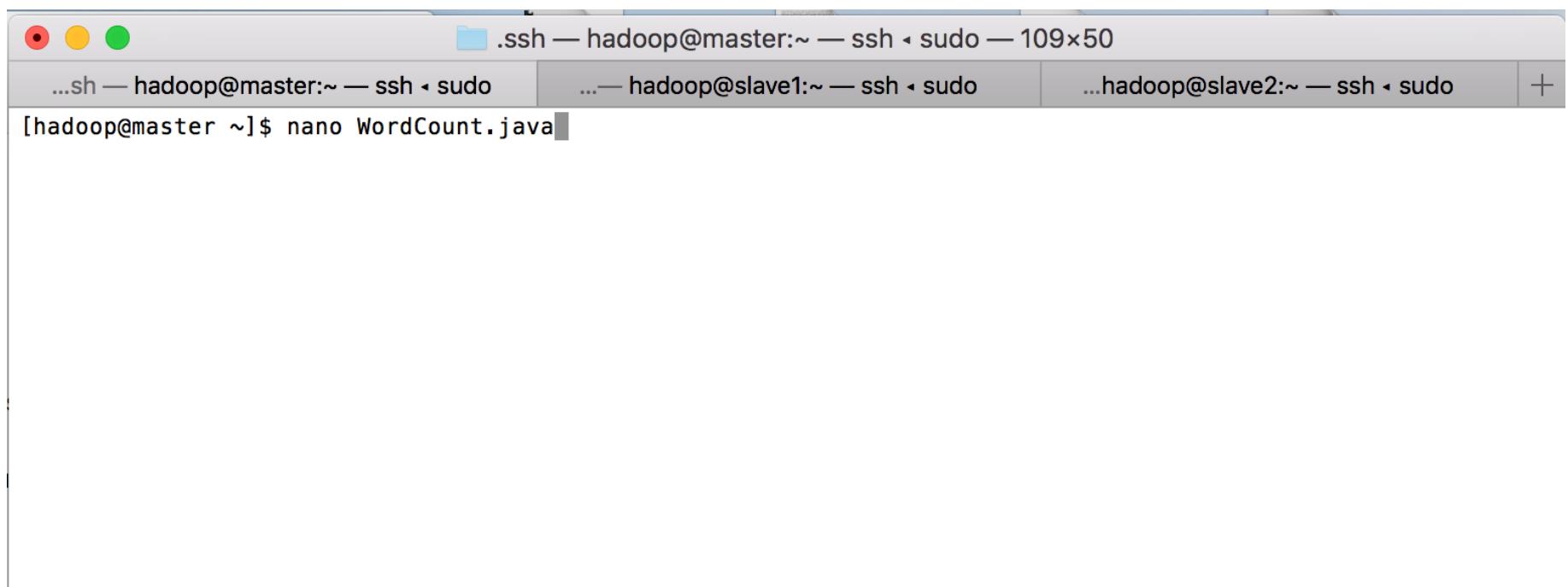
The screenshot shows the 'Launch Instance' configuration page with the following sections:

- Instance Type:** t2.micro (ECUs: Variable, vCPUs: 1, Memory: 1 GiB, Instance Storage: EBS only, EBS-Optimized Available: -, Network Performance: Low to Moderate)
- Security Groups:** (Edit security groups)
- Instance Details:** (Edit instance details)
- Storage:** (Edit storage)

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/xvda	snapshot-0c44a2efc327e7ee6	8	gp2	100 / 3000	N/A	Yes	Not Encrypted

Word Count Problem -MapReduce Program

Create a WordCount java program.



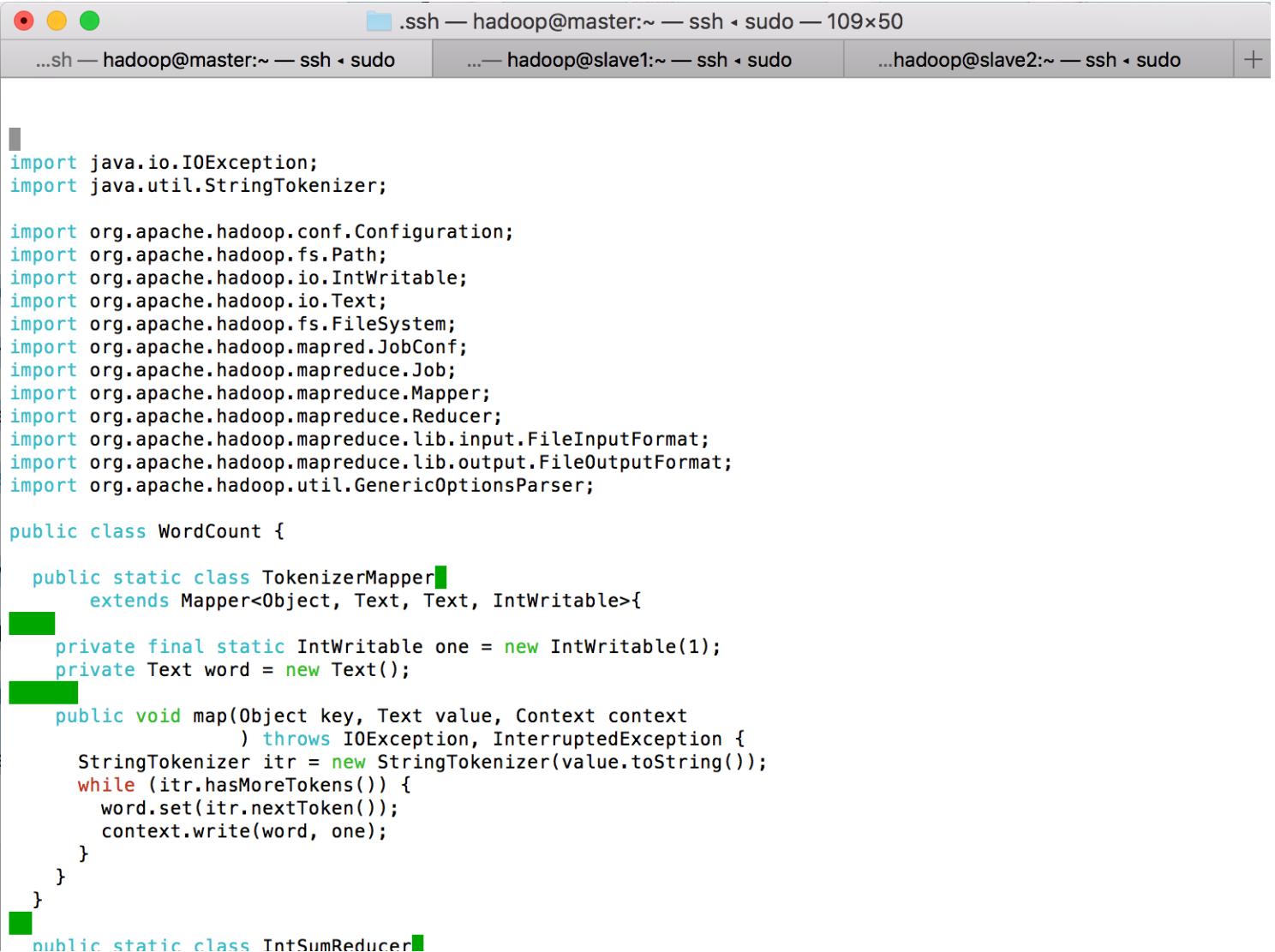
The screenshot shows a terminal window with the following details:

- Window title: .ssh — hadoop@master:~ — ssh ▾ sudo — 109x50
- Tab bar:
 - ...sh — hadoop@master:~ — ssh ▾ sudo
 - ...— hadoop@slave1:~ — ssh ▾ sudo
 - ...hadoop@slave2:~ — ssh ▾ sudo
 - +
- Terminal content:

```
[hadoop@master ~]$ nano WordCount.java
```

Word Count Problem -MapReduce Program

Mapper Function:



The screenshot shows a terminal window titled ".ssh — hadoop@master:~ — ssh - sudo — 109x50". The window has three tabs: "...sh — hadoop@master:~ — ssh - sudo", "...— hadoop@slave1:~ — ssh - sudo", and "...hadoop@slave2:~ — ssh - sudo". The main pane displays Java code for a MapReduce WordCount program. The code includes imports for various Hadoop classes and defines a WordCount class with a TokenizerMapper and a TntSumReducer.

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;

public class WordCount {

    public static class TokenizerMapper
        extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context
                       ) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class TntSumReducer
        extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();
}
```

Word Count Problem -MapReduce Program

Reducer Function:

```
public static class IntSumReducer
    extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable> values,
                      Context context
                      ) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}
```

Word Count Problem -MapReduce Program

Main Function:

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    String[] otherArgs = new GenericOptionsParser(conf, args).getRemainingArgs();
    if (otherArgs.length != 2) {
        System.err.println("Usage: wordcount <in> <out>");
        System.exit(2);
    }

    FileSystem hdfs = FileSystem.get(conf);
    Path findf=new Path(otherArgs[1]);
    boolean isExists=hdfs.exists(findf);
    System.out.println("exit?"+isExists);
    if(isExists)
    {
        hdfs.delete(findf, true);
        System.out.println("delete output");
    }
}

Job job = new Job(conf, "word count");
job.setJarByClass(WordCount.class);
job.setMapperClass(TokenizerMapper.class);
job.setCombinerClass(IntSumReducer.class);
job.setReducerClass(IntSumReducer.class);
job.setOutputKeyClass(Text.class);
job.setOutputValueClass(IntWritable.class);
FileInputFormat.addInputPath(job, new Path(otherArgs[0]));
FileOutputFormat.setOutputPath(job, new Path(otherArgs[1]));
System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

Word Count Problem

-Compile

Compile.

```
[[hadoop@master ~]$ nano WordCount.java
[[hadoop@master ~]$ javac WordCount.java -cp $(hadoop classpath)
Note: WordCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[hadoop@master ~]$ ]]
```

Word Count Problem

-Compile

Export JAR.

The screenshot shows a terminal window with four tabs. The active tab is titled '.ssh — hadoop@master:~ — ssh ▾ sudo — 109x50'. The other tabs are labeled '@master:~ — ssh ▾ sudo', '~/.ssh — -bash', and '...0/subdir0 — ssh ▾ sudo' and '.../subdir0 — ssh ▾ sudo'. The terminal output is as follows:

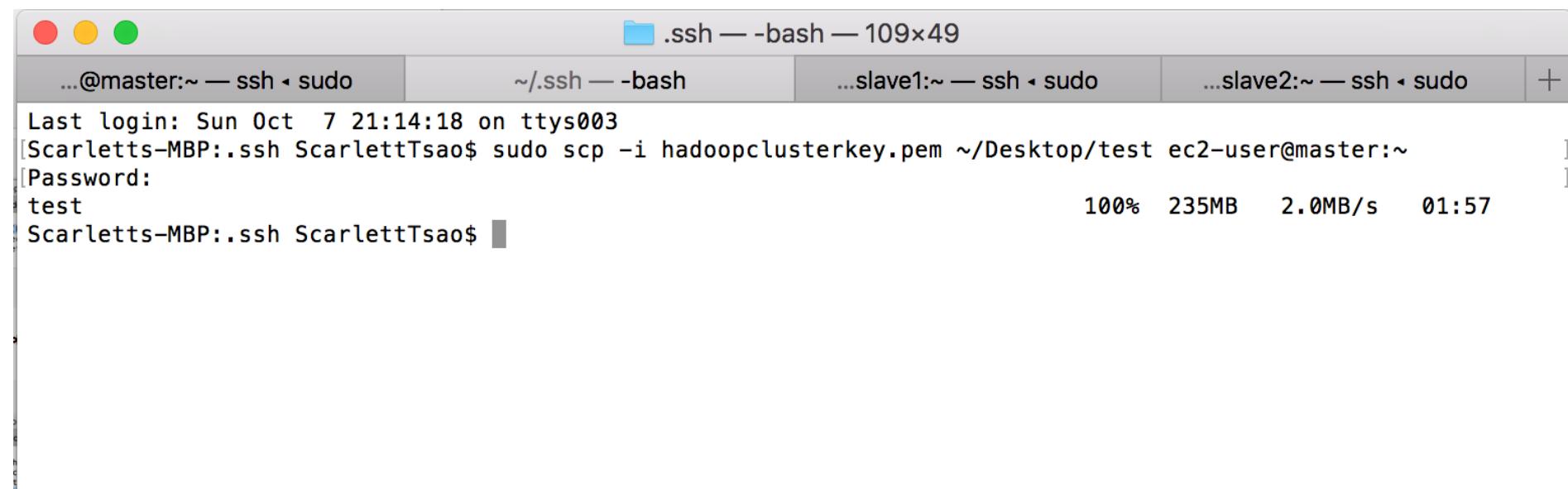
```
[hadoop@master ~]$ nano WordCount.java
[hadoop@master ~]$ javac WordCount.java -cp $(hadoop classpath)
Note: WordCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[hadoop@master ~]$ jar -cvf WordCount.jar ./WordCount*.class
added manifest
adding: WordCount.class(in = 2412) (out= 1251)(deflated 48%)
adding: WordCount$IntSumReducer.class(in = 1739) (out= 741)(deflated 57%)
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 753)(deflated 56%)
```

Word Count Problem

-Input

Upload the input data from local to the master node.

The input size is 235MB.



A screenshot of a macOS terminal window titled ".ssh — -bash — 109x49". The window has four tabs: "...@master:~ — ssh < sudo", "~/.ssh — -bash", "...slave1:~ — ssh < sudo", and "...slave2:~ — ssh < sudo". The main pane shows the command:

```
Last login: Sun Oct  7 21:14:18 on ttys003
[Scarletts-MBP:.ssh ScarlettTsao$ sudo scp -i hadoopclusterkey.pem ~/Desktop/test ec2-user@master:~
>Password:
test
Scarletts-MBP:.ssh ScarlettTsao$
```

At the bottom right, there is a progress bar indicating "100% 235MB 2.0MB/s 01:57".

Word Count Problem

-Input

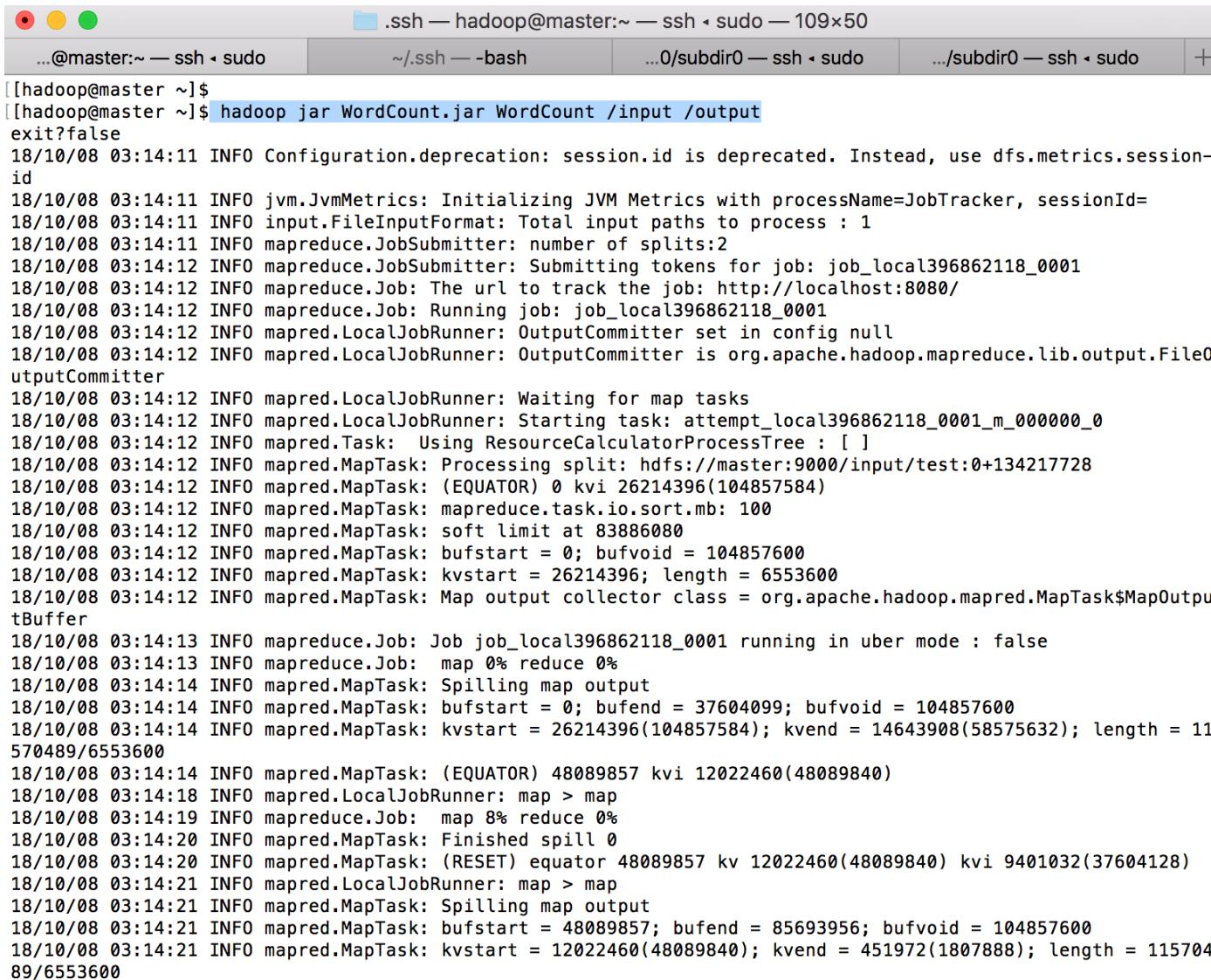
Create the input directory in HDFS and place the input file in it.

```
[[hadoop@master ~]$ nano WordCount.java
[[hadoop@master ~]$ javac WordCount.java -cp $(hadoop classpath)
Note: WordCount.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
[[hadoop@master ~]$ jar -cvf WordCount.jar ./WordCount*.class
added manifest
adding: WordCount.class(in = 2412) (out= 1251)(deflated 48%)
adding: WordCount$IntSumReducer.class(in = 1739) (out= 741)(deflated 57%)
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 753)(deflated 56%)
[[hadoop@master ~]$ hadoop fs -mkdir /input
[[hadoop@master ~]$ exit
exit
[[ec2-user@ip-10-0-0-120 ~]$ ls
test
[[ec2-user@ip-10-0-0-120 ~]$ sudo mv test /home/hadoop
[[ec2-user@ip-10-0-0-120 ~]$ su hadoop
[Password:
[[hadoop@master ec2-user]$ cd
[[hadoop@master ~]$ source /etc/profile
[[hadoop@master ~]$ jps
3473 SecondaryNameNode
4411 Jps
3294 NameNode
3615 ResourceManager
[[hadoop@master ~]$ hadoop fs -put test /input
[[hadoop@master ~]$
```

Word Count Problem

-Execution

Execute JAR file on the cluster.



The screenshot shows a terminal window with four tabs. The active tab displays the command and its output:

```
.ssh — hadoop@master:~ — ssh - sudo — 109x50
...@master:~ — ssh - sudo      ~/ssh — -bash      ...0/subdir0 — ssh - sudo      .../subdir0 — ssh - sudo      +
```

```
[[hadoop@master ~]$  
[[hadoop@master ~]$ hadoop jar WordCount.jar WordCount /input /output  
exit?false  
18/10/08 03:14:11 INFO Configuration.deprecation: session.id is deprecated. Instead, use dfs.metrics.session-id  
18/10/08 03:14:11 INFO jvm.JvmMetrics: Initializing JVM Metrics with processName=JobTracker, sessionId=  
18/10/08 03:14:11 INFO input.FileInputFormat: Total input paths to process : 1  
18/10/08 03:14:11 INFO mapreduce.JobSubmitter: number of splits:2  
18/10/08 03:14:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local396862118_0001  
18/10/08 03:14:12 INFO mapreduce.Job: The url to track the job: http://localhost:8080/  
18/10/08 03:14:12 INFO mapreduce.Job: Running job: job_local396862118_0001  
18/10/08 03:14:12 INFO mapred.LocalJobRunner: OutputCommitter set in config null  
18/10/08 03:14:12 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.lib.output.FileOutputCommitter  
18/10/08 03:14:12 INFO mapred.LocalJobRunner: Waiting for map tasks  
18/10/08 03:14:12 INFO mapred.LocalJobRunner: Starting task: attempt_local396862118_0001_m_000000_0  
18/10/08 03:14:12 INFO mapred.Task: Using ResourceCalculatorProcessTree : []  
18/10/08 03:14:12 INFO mapred.MapTask: Processing split: hdfs://master:9000/input/test:0+134217728  
18/10/08 03:14:12 INFO mapred.MapTask: (EQUATOR) 0 kvi 26214396(104857584)  
18/10/08 03:14:12 INFO mapred.MapTask: mapreduce.task.io.sort.mb: 100  
18/10/08 03:14:12 INFO mapred.MapTask: soft limit at 83886080  
18/10/08 03:14:12 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600  
18/10/08 03:14:12 INFO mapred.MapTask: kvstart = 26214396; length = 6553600  
18/10/08 03:14:12 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer  
18/10/08 03:14:13 INFO mapreduce.Job: Job job_local396862118_0001 running in uber mode : false  
18/10/08 03:14:13 INFO mapreduce.Job: map 0% reduce 0%  
18/10/08 03:14:14 INFO mapred.MapTask: Spilling map output  
18/10/08 03:14:14 INFO mapred.MapTask: bufstart = 0; bufend = 37604099; bufvoid = 104857600  
18/10/08 03:14:14 INFO mapred.MapTask: kvstart = 26214396(104857584); kvend = 14643908(58575632); length = 11570489/6553600  
18/10/08 03:14:14 INFO mapred.MapTask: (EQUATOR) 48089857 kvi 12022460(48089840)  
18/10/08 03:14:18 INFO mapred.LocalJobRunner: map > map  
18/10/08 03:14:19 INFO mapreduce.Job: map 8% reduce 0%  
18/10/08 03:14:20 INFO mapred.MapTask: Finished spill 0  
18/10/08 03:14:20 INFO mapred.MapTask: (RESET) equator 48089857 kv 12022460(48089840) kvi 9401032(37604128)  
18/10/08 03:14:21 INFO mapred.LocalJobRunner: map > map  
18/10/08 03:14:21 INFO mapred.MapTask: Spilling map output  
18/10/08 03:14:21 INFO mapred.MapTask: bufstart = 48089857; bufend = 85693956; bufvoid = 104857600  
18/10/08 03:14:21 INFO mapred.MapTask: kvstart = 12022460(48089840); kvend = 451972(1807888); length = 11570489/6553600
```

Word Count Problem

-Execution

Execute successfully.

```
...@master:~ — ssh ✘ sudo          ~./ssh — -bash          ...slave1:~ — ssh ✘ sudo          ...slave2:~ — ssh ✘ sudo          +  
18/10/08 03:15:56 INFO mapred.Task: Task 'attempt_local396862118_0001_r_000000_0' done.  
18/10/08 03:15:56 INFO mapred.LocalJobRunner: Finishing task: attempt_local396862118_0001_r_000000_0  
18/10/08 03:15:56 INFO mapred.LocalJobRunner: reduce task executor complete.  
18/10/08 03:15:57 INFO mapreduce.Job: map 100% reduce 100%  
18/10/08 03:15:57 INFO mapreduce.Job: Job job_local396862118_0001 completed successfully  
18/10/08 03:15:57 INFO mapreduce.Job: Counters: 38  
    File System Counters  
        FILE: Number of bytes read=1866215935  
        FILE: Number of bytes written=2504524456  
        FILE: Number of read operations=0  
        FILE: Number of large read operations=0  
        FILE: Number of write operations=0  
        HDFS: Number of bytes read=626982338  
        HDFS: Number of bytes written=301126419  
        HDFS: Number of read operations=28  
        HDFS: Number of large read operations=0  
        HDFS: Number of write operations=5  
    Map-Reduce Framework  
        Map input records=27375129  
        Map output records=27375129  
        Map output bytes=355876677  
        Map output materialized bytes=410626947  
        Input split bytes=188  
        Combine input records=54750258  
        Combine output records=54750258  
        Reduce input groups=27375129  
        Reduce shuffle bytes=410626947  
        Reduce input records=27375129  
        Reduce output records=27375129  
        Spilled Records=82125387  
        Shuffled Maps =2  
        Failed Shuffles=0  
        Merged Map outputs=2  
        GC time elapsed (ms)=1318  
        CPU time spent (ms)=0  
        Physical memory (bytes) snapshot=0  
        Virtual memory (bytes) snapshot=0  
        Total committed heap usage (bytes)=457912320  
    Shuffle Errors  
        BAD_ID=0  
        CONNECTION=0  
        IO_ERROR=0  
        WRONG_LENGTH=0  
        WRONG_MAP=0  
        WRONG_REDUCE=0  
    File Input Format Counters  
        Bytes Read=246380257  
    File Output Format Counters  
        Bytes Written=301126419  
[hadoop@master ~]$
```

Word Count Problem

-Output

Finds out the size of the output, which is 287MB.

```
Bytes Written=301126419
[[hadoop@master ~]$ hadoop fs -ls /output
Found 2 items
-rw-r--r--    1 hadoop supergroup      0 2018-10-08 03:15 /output/_SUCCESS
-rw-r--r--    1 hadoop supergroup  301126419 2018-10-08 03:15 /output/part-r-00000
[[hadoop@master ~]$ hadoop dfsadmin -report
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.
```

```
3692 ResourceManager
[hadoop@ip-10-0-0-120 ec2-user]$ hadoop fsck / -files -blocks -locations
[DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Connecting to namenode via http://master:50070
FSCK started by hadoop (auth:SIMPLE) from /10.0.0.120 for path / at Mon Oct 08 21:53:06
UTC 2018
/ <dir>
/input <dir>
/input/test 246376161 bytes, 2 block(s): OK
0. BP-676360153-10.0.0.120-1538963995663:blk_1073741825_1001 len=134217728 repl=1 [10.0
.0.76:50010]
1. BP-676360153-10.0.0.120-1538963995663:blk_1073741826_1002 len=112158433 repl=1 [10.0
.0.76:50010]

/output <dir>
/output/_SUCCESS 0 bytes, 0 block(s): OK

/output/part-r-00000 301126419 bytes, 3 block(s): OK
0. BP-676360153-10.0.0.120-1538963995663:blk_1073741827_1003 len=134217728 repl=1 [10.0
.0.157:50010]
1. BP-676360153-10.0.0.120-1538963995663:blk_1073741828_1004 len=134217728 repl=1 [10.0
.0.76:50010]
2. BP-676360153-10.0.0.120-1538963995663:blk_1073741829_1005 len=32690963 repl=1 [10.0
.0.157:50010]

Status: HEALTHY
Total size:      547502580 B
Total dirs:     3
Total files:    3
Total symlinks:          0
Total blocks (validated):   5 (avg. block size 109500516 B)
Minimally replicated blocks: 5 (100.0 %)
Over-replicated blocks:    0 (0.0 %)
Under-replicated blocks:   0 (0.0 %)
Mis-replicated blocks:    0 (0.0 %)
Default replication factor: 1
Average block replication: 1.0
Corrupt blocks:           0
Missing replicas:          0 (0.0 %)
Number of data-nodes:      2
Number of racks:           1
FSCK ended at Mon Oct 08 21:53:06 UTC 2018 in 9 milliseconds

The filesystem under path '/' is HEALTHY
[hadoop@ip-10-0-0-120 ec2-user]$
```

Word Count Problem

-Output Location

Two datablocks on slave 1.
(160.42MB)

Three datablocks on slave 2.
(365.80MB)

```
...@master:~ — ssh • sudo      .ssh — hadoop@master:~ — ssh • sudo — 109x50
...@master:~ — ssh • sudo      ~/ssh — bash           ...slave1:~ — ssh • sudo    ...slave2:~ — ssh • sudo
[hadoop@master ~]$ hadoop dfsadmin -report
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Configured Capacity: 17154662400 (15.98 GB)
Present Capacity: 13287440015 (12.37 GB)
DFS Remaining: 12735651840 (11.86 GB)
DFS Used: 551788175 (526.23 MB)
DFS Used%: 4.15%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

-----
Live datanodes (2):

Name: 10.0.0.157:50010 (slave1)
Hostname: slave1
Decommission Status : Normal
Configured Capacity: 8577331200 (7.99 GB)
DFS Used: 168216777 (160.42 MB)
Non DFS Used: 1933563703 (1.80 GB)
DFS Remaining: 6475550720 (6.03 GB)
DFS Used%: 1.96%
DFS Remaining%: 75.50%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon Oct 08 03:18:05 UTC 2018

Name: 10.0.0.76:50010 (slave2)
Hostname: slave2
Decommission Status : Normal
Configured Capacity: 8577331200 (7.99 GB)
DFS Used: 383571398 (365.80 MB)
Non DFS Used: 1933658682 (1.80 GB)
DFS Remaining: 6260101120 (5.83 GB)
DFS Used%: 4.47%
DFS Remaining%: 72.98%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Mon Oct 08 03:18:05 UTC 2018
```

Word Count Problem

-Output Location

Check slave node 1: There are two data blocks stored on this node. The data block size is 128MB.

By checking the first 10 lines of the two datablocks' contents, we see that both datablocks stored on slave node 1 are the output.

```
ssh — hadoop@slave1:~/hadoop-2.6.0/data/current/BP-676360153-10.0.0.120-1538963995663/current/finalize...
...@master:~ — ssh + sudo          ~/.ssh — bash           ...0/subdir0 — ssh + sudo    ...slave2:~ — ssh + sudo + 
[[hadoop@slave1 ~]$ cd hadoop-2.6.0/
[[hadoop@slave1 hadoop-2.6.0]$ ls
bin  data  etc  include  lib  libexec  LICENSE.txt  logs  NOTICE.txt  README.txt  sbin  share  tmp
[[hadoop@slave1 hadoop-2.6.0]$ cd data
[[hadoop@slave1 data]$ ls
current  in_use.lock
[[hadoop@slave1 data]$ du -m *
161      current/BP-676360153-10.0.0.120-1538963995663/current/finalized/subdir0/subdir0
161      current/BP-676360153-10.0.0.120-1538963995663/current/finalized/subdir0
161      current/BP-676360153-10.0.0.120-1538963995663/current/finalized
0       current/BP-676360153-10.0.0.120-1538963995663/current/rbw
161      current/BP-676360153-10.0.0.120-1538963995663/current
0       current/BP-676360153-10.0.0.120-1538963995663/tmp
161      current/BP-676360153-10.0.0.120-1538963995663
161      current
1       in_use.lock
[[hadoop@slave1 data]$ cd current/BP-676360153-10.0.0.120-1538963995663/current/finalized/subdir0/subdir0
[[hadoop@slave1 subdir0]$ ls
blk_1073741827  blk_1073741827_1003.meta  blk_1073741829  blk_1073741829_1005.meta
[[hadoop@slave1 subdir0]$ head -10 blk_1073741827
aaaaaaaa      1
aaaaaaaaab     1
aaaaaaaaac     1
aaaaaaaaaba   1
aaaaaaaaabb   1
aaaaaaaaabc   1
aaaaaaaaaca   1
aaaaaaaaacb   1
aaaaaaaaacc   1
aaaaaaaaada   1
[[hadoop@slave1 subdir0]$ head -10 blk_1073741829
uvdvc      1
aaruvdwa     1
aaruvdwdb    1
aaruvdwc    1
aaruvdxsa   1
aaruvdxb    1
aaruvdxc    1
aaruvdyaa   1
aaruvdyb    1
aaruvdycc   1
[[hadoop@slave1 subdir0]$
```

Word Count Problem -Output Location

Check slave node 2: There are three data blocks stored on this node. The data block size is 128MB.

By checking the first 10 lines of the three datablocks' contents, we see that the third datablock stored on slave node 2 is the output, while the other two store the keys.

```
○ ○ ○ .ssh — hadoop@slave2:~/hadoop-2.6.0/data/current/BP-676360153-10.0.0.120-1538963995663/current/finaliz...
...@master:~ — ssh + sudo      ~/ssh — bash      .../subdir0 — ssh + sudo      .../subdir0 — ssh + sudo      +
[[hadoop@slave2 ~]$ cd hadoop-2.6.0/
[[hadoop@slave2 hadoop-2.6.0]$ ls
bin  data  etc  include  lib  libexec  LICENSE.txt  logs  NOTICE.txt  README.txt  sbin  share  tmp
[[hadoop@slave2 hadoop-2.6.0]$ cd data
[[hadoop@slave2 data]$ ls
current  in_use.lock
[[hadoop@slave2 data]$ du -m *
366    current/BP-676360153-10.0.0.120-1538963995663/current/finalized/subdir0/subdir0
366    current/BP-676360153-10.0.0.120-1538963995663/current/finalized/subdir0
366    current/BP-676360153-10.0.0.120-1538963995663/current/finalized
0     current/BP-676360153-10.0.0.120-1538963995663/current/rbw
366    current/BP-676360153-10.0.0.120-1538963995663/current
0     current/BP-676360153-10.0.0.120-1538963995663/tmp
366    current/BP-676360153-10.0.0.120-1538963995663
366    current
1     in_use.lock
[[hadoop@slave2 data]$ cd current/BP-676360153-10.0.0.120-1538963995663/current/finalized/subdir0/subdir0
[[hadoop@slave2 subdir0]$ ls
blk_1073741825          blk_1073741826          blk_1073741828
blk_1073741825_1001.meta  blk_1073741826_1002.meta  blk_1073741828_1004.meta
[[hadoop@slave2 subdir0]$ head -10 blk_1073741825
aaaaaaaaaa
aaaaaaaaab
aaaaaaaaac
aaaaaaaaab
aaaaaaaaabb
aaaaaaaaabc
aaaaaaaaaca
aaaaaaaaacb
aaaaaaaaacc
aaaaaaaaada
[[hadoop@slave2 subdir0]$ head -10 blk_1073741826
aakwvpja
aakwvpjb
aakwvpjc
aakwvpka
aakwvpkb
aakwvpkc
aakwvpbla
aakwvpblb
aakwvplc
[[hadoop@slave2 subdir0]$ head -10 blk_1073741828
c      1
aaixkoya      1
aaixkoyb      1
aaixkoyc      1
aaixkoza      1
aaixkozb      1
```

Hadoop configuration

□ Standalone Mode

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SingleCluster.html#Configuration>

□ Pseudo-Distributed Mode

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SingleCluster.html#Pseudo-Distributed_Operation

□ Fully-Distributed Mode

<https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/ClusterSetup.html>

How to install Hadoop on Amazon AWS (step by step):

<https://www.youtube.com/watch?v=a-DXDkK1i08>

Another useful tutorial:

<https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster>

Java programming

How to MapReduce programming with Apache:

<https://www.javaworld.com/article/2077907/open-source-tools/mapreduce-programming-with-apache-hadoop.html>

If you need more details, the following book helps:

<https://eecs.wsu.edu/~yinghui/mat/courses/fall%202015/resources/Hadoop%20the%20definitive%20guide.pdf>

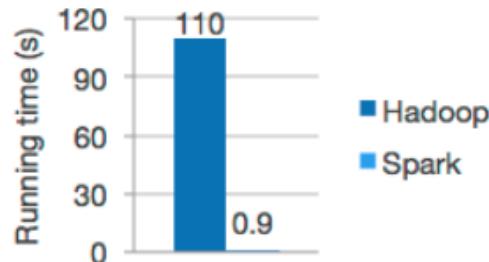
The following tutorial shows you how to use Eclipse to write, compile, execute and export .jar file for the word counting problem in Hadoop in detail:

<https://www.dezyre.com/hadoop-tutorial/hadoop-mapreduce-wordcount-tutorial>

Speed

Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.



Logistic regression in Hadoop and Spark

Ease of Use

Write applications quickly in Java, Scala or Python.

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it *interactively* from the Scala and Python shells.

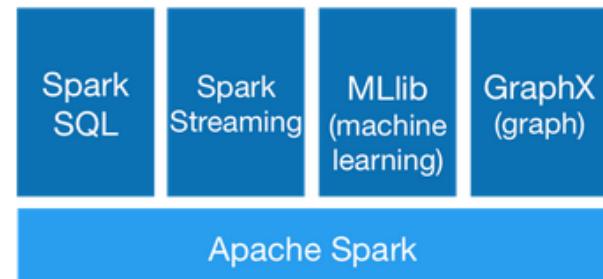
```
file = spark.textFile("hdfs://...")  
  
file.flatMap(lambda line: line.split())  
.map(lambda word: (word, 1))  
.reduceByKey(lambda a, b: a+b)
```

Word count in Spark's Python API

Generality

Combine SQL, streaming, and complex analytics.

Spark powers a stack of high-level tools including [Spark SQL](#), [MLlib](#) for machine learning, [GraphX](#), and [Spark Streaming](#). You can combine these frameworks seamlessly in the same application.



Big Data Visualization

- Graph Database
- Visual Analytics

76,425 species



Tree of Life by Dr. Yifan Hu

14.8 million tweets



The information diffusion graph of the death of Osama bin Laden by Gilad Lotan

500 million users

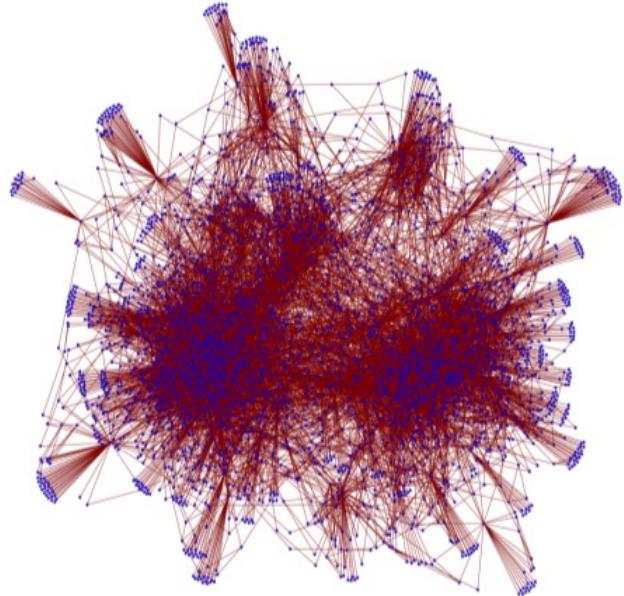


Facebook friendship graph by Paul Butler

Challenging Task :

Squeezing millions and even billions of records into million pixels ($1600 \times 1200 \approx 2$ million pixels)

Visualization Key Challenges



Visual clutter

How can we encode the information intuitively?



Performance issues

How can we render the huge datasets in real time with rich interactions?

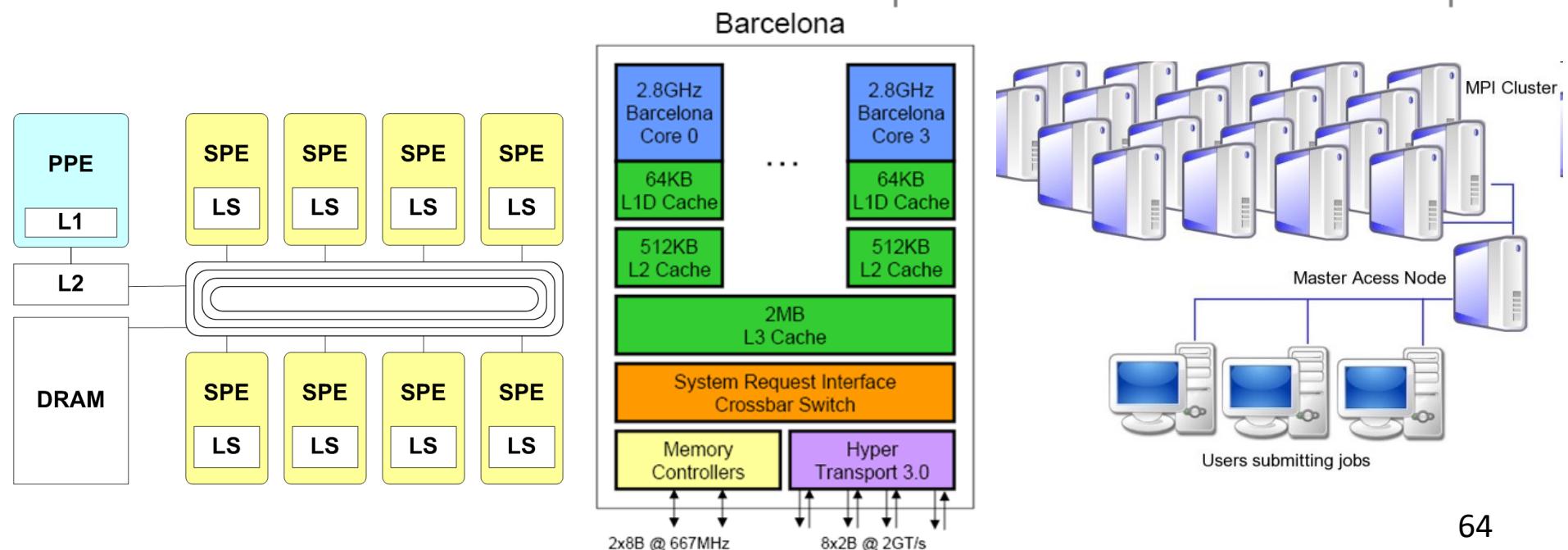
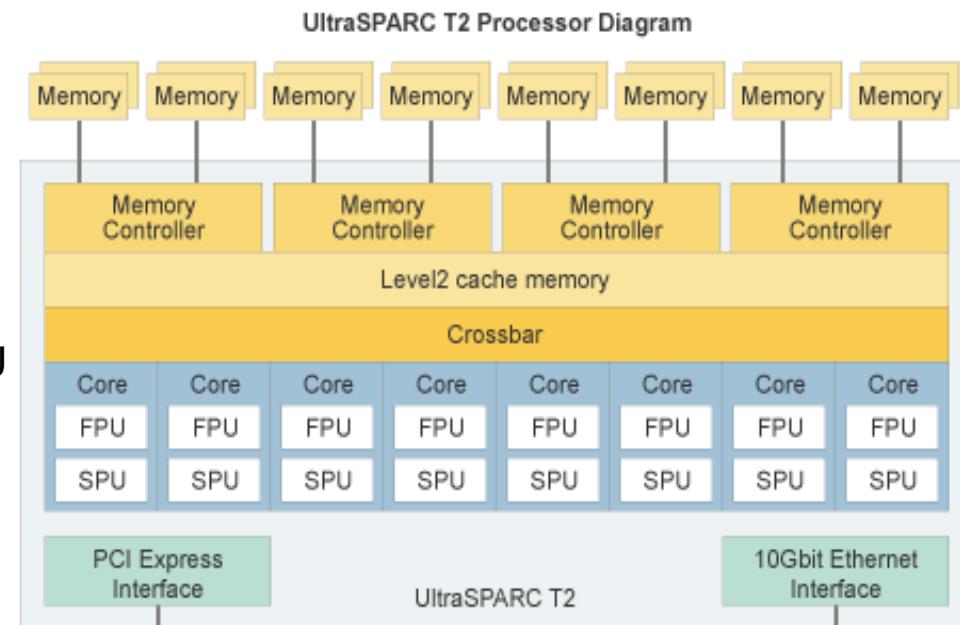


Cognition

How can users understand the visual representation when the information is overwhelming?

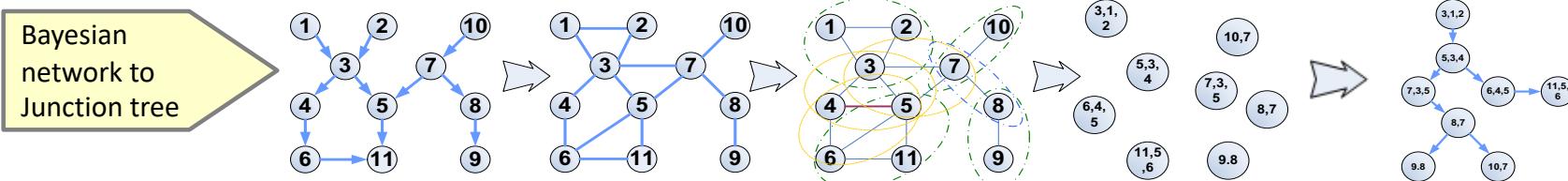
Platform Dependent Graphical Models

- Homogeneous **multicore** processors
 - Intel Xeon E5335 (Clovertown)
 - AMD Opteron 2347 (Barcelona)
 - Netezza (FPGA, multicore)*
- Homogeneous **manycore** processors
 - Sun UltraSPARC T2 (Niagara 2), GPGPU
- **Heterogeneous** multicore processors
 - Cell Broadband Engine
- **Clusters**
 - HPCC, DataStar, *BlueGene*, etc.



Graph Workload Types

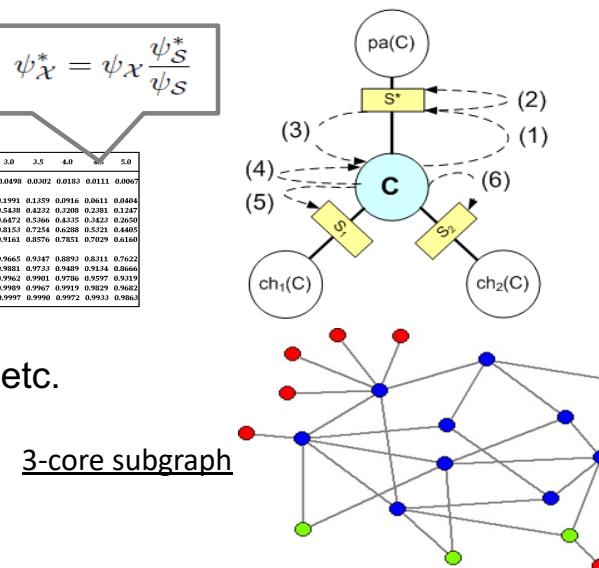
- Type 1: Computations on graph structures / topologies
 - Example → converting Bayesian network into junction tree, **graph traversal (BFS/DFS)**, etc.
 - Characteristics → Poor locality, irregular memory access, limited numeric operations



- Type 2: Computations on graphs with rich properties
 - Example → **Belief propagation**: diffuse information through a graph using statistical models
 - Characteristics
 - Locality and memory access pattern depend on vertex models
 - Typically a lot of numeric operations
 - Hybrid workload
- Type 3: Computations on dynamic graphs
 - Example → **streaming graph clustering**, incremental k-core, etc.
 - Characteristics
 - Poor locality, irregular memory access
 - Operations to update a model (e.g., cluster, sub-graph)
 - Hybrid workload

$$\psi_{\mathcal{S}}^* = \sum_{\mathcal{Y} \setminus \mathcal{S}} \psi_{\mathcal{Y}}^*, \quad \psi_{\mathcal{X}}^* = \psi_{\mathcal{X}} \frac{\psi_{\mathcal{S}}^*}{\psi_{\mathcal{S}}}$$

λ	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
X=0	0.6005	0.3679	0.2231	0.1353	0.0821	0.0439	0.0302	0.0183	0.0114	0.0067
1	0.9998	0.7358	0.5578	0.4660	0.2873	0.1991	0.1359	0.0916	0.0611	0.0404
2	0.9858	0.6197	0.5197	0.4800	0.2767	0.1548	0.1222	0.0808	0.0531	0.0347
3	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
4	0.9998	0.9983	0.9814	0.9473	0.8912	0.8153	0.7254	0.6288	0.5321	0.4405
5	1.0000	0.9994	0.9994	0.9955	0.9834	0.9161	0.8576	0.7851	0.7029	0.6160
6	1.0000	0.9999	0.9991	0.9955	0.9838	0.9665	0.9447	0.8891	0.8311	0.7622
7	1.0000	1.0000	0.9999	0.9989	0.9958	0.9881	0.9733	0.9489	0.9134	0.8666
8	1.0000	1.0000	1.0000	0.9998	0.9989	0.9962	0.9901	0.9796	0.9597	0.9119
9	1.0000	1.0000	1.0000	1.0000	0.9999	0.9999	0.9999	0.9999	0.9993	0.9863
10	1.0000	1.0000	1.0000	1.0000	1.0000	0.9997	0.9999	0.9992	0.9933	0.9863



Large-scale graph benchmark – Graph 500

June 2024, Breadth-First Search (GTEPS)

RANK	PREVIOUS RANK	MACHINE	VENDOR	TYPE	NETWORK	INSTALLATION SITE	LOCATION	COUNTRY	YEAR	APPLICATION	USAGE	NUMBER OF NODES	NUMBER OF CORES	MEMORY	IMPLEMENTATION	SCALE	GTEPS	C_TIME	POWER
1	new	Supercomputer Fugaku	Fujitsu	Fujitsu A64FX	Tofu Interconnect D	RIKEN Center for Computational Science (R-CCS)	Kobe Hyogo	Japan	2020	Various scientific and industrial fields	Academic and industry	152064	7299072	4866048	Custom	42	166029	4350.38	
2	2	Wuhan Supercomputer	HUST	Kunpeng 920+Tesla A100	Custom	Wuhan Supercomputing Center	Wuhan	China	2023	Research	Academic	252	6999552	584640	Custom (Graph Processing System "YITU" implemented by Jin Zhao Yu Zhang et al.)	41	115357.6	16284.8	
3	3	Frontier	HPE	HPE Cray EX235a	Slingshot-11	DOE/SC/Oak Ridge National Laboratory	Oak Ridge TN	United States	2021	government	open research	9248	8730112	9469952	Fugaku Derivative	40	29654.6	29.7099	NA
4	4	Pengcheng Cloudbrain-II	HUST-Pengcheng Lab-HUAWEI	Kunpeng 920+Ascend 910	Custom	Pengcheng Lab	ShenZhen	China	2022	Research	Academic	488	93696	999424	Custom (Graph Processing System "YITU" implemented by Jin Zhao Yu Jian Liao Yu Zhang et al.)	40	28463.1	57218.3	
5	new	Aurora	Intel/HPE	HPE Cray EX - Intel Exascale Compute Blade	HPE Slingshot-11	DOE/SC/Argonne National Laboratory	Argonne IL	United States	2023	Open research	government	4096	25591808	4718592	Custom "Fugaku Derivative"	40	24250.2	240.394	
6	5	Sunway TaihuLight	NRPC	Sunway MPP	Sunway	National Supercomputing Center in Wuxi	Wuxi	China	2015	research	research	40768	10599680	1304580 gigabytes	Custom	40	23755.7	961.455 seconds	
7	6	Wisteria/BDEC-01 (Odyssey)	Fujitsu	PRIMEHPC FX1000	Tofu interconnect D	Information Technology Center The University of Tokyo	Kashiwa Chiba	Japan	2021	University	Research	7680	368640	245760	Custom	37	16118	90.7395	
8	new	MareNostrum 5 ACC	Eviden	BullSequana XH3000 XEON PLATINUM 8460Y+ 40C 2.3GHZ NVIDIA H100 64GB INFINIBAND NDR200	NDR200	Barcelona Supercomputing Center	Barcelona	Spain	2024	Research and innovation	all open R&D	1120	680960	573400	BFS2D-NVIDIA	35	15737.43	86.65	
9	7	TOKI-SORA	Fujitsu	PRIMEHPC FX1000	Tofu interconnect D	Japan Aerospace Exploration Agency (JAXA)	Tokyo	Japan	2020	Research	CFD	5760	276480	184320	Custom	36	10813	70.5226	
10	8	NAPS-FX1000	Fujitsu	PRIMEHPC FX1000	Tofu interconnect D	Japan Meteorological Agency	Tokyo	Japan	2022	Research	Weather and Climate Research	4608	221184	147456	Custom	36	10158	78.5127	

Common Use Cases for Big Data in Hadoop

- Log Data Analysis
 - most common, fits perfectly for HDFS scenario: Write once & Read often.
- Data Warehouse Modernization
- Fraud Detection
- Risk Modeling
- Social Sentiment Analysis
- Image Classification
- Graph Analysis
- Beyond

Big Data Analytics Example Use Cases

1. Social Network Analysis
2. Recommendation
3. Commerce
4. Financial Analysis
5. Social Media Monitoring
6. Telco Customer Analysis
7. Watson
8. Data Exploration and Visualization
9. Personalized Search
10. Anomaly Detection (Espionage, Sabotage, etc.)
11. Fraud Detection
12. Cybersecurity
13. Sensor Monitoring (Smarter another Planet)
14. Cellular Network Monitoring
15. Cloud Monitoring
16. Code Life Cycle Management
17. Traffic Navigation
18. Image and Video Semantic Understanding
19. Genomic Medicine
20. Brain Network Analysis
21. Data Curation
22. Near Earth Object Analysis



Use Case 1: Social Network Analysis in Enterprise for Productivity

Production Live System used by IBM GBS since 2009 – verified ~\$100M contribution

15,000 contributors in 76 countries; 92,000 annual unique IBM users

25,000,000+ emails & SameTime messages (incl. Content features)

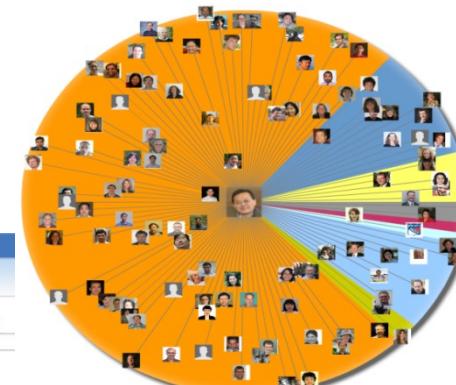
1,000,000+ Learning clicks; 14M KnowledgeView, SalesOne, ..., access data

1,000,000+ Lotus Connections (blogs, file sharing, bookmark) data

200,000 people's consulting project & earning data

The screenshot shows the SmallBlue Suite interface with a search bar for 'subject keywords' set to 'healthcare'. Below the search bar, there are dropdown menus for 'Country' (all) and 'Division' (Advanced search). A 'Find Experts' button is visible. To the right, a network visualization titled 'SmallBlue Net' shows a cluster of nodes connected by lines. On the left, a list of six search results is displayed:

1. Patricia (Patti) Okita: Global Business Services Associate Partner, Healthcare Integration Other Consultant Ask: MARTHA E. (Martha) GIBSON > Amy D. (AMY) Berk
2. Michael Hohenberger: IBM Research Life Sciences Business Development Category Sales Ask: Ravi B. Konuru > Vanessa L.
3. Todd (T.H.) Kalynuk: Global Business Services GBS Partner, Healthcare and Public Health -- Practice Administrator is Shirley Carkner Other Consultant Ask: Chung Sheng Li > Robert (R.) Torek
4. Susan E. (SUSAN) Rivers: Global Business Services Healthcare Knowledge Manager Market Insights Ask: MARTHA E. (Martha) GIBSON
5. M.C. (Mark) Effingham: IBM Research - Almaden Research Center
6. Paul (P.E.) Van Aggelen: Global Business Services



Shortest Paths

Centralities

Graph Search

The screenshot shows the SmallBlue Suite interface with a 'Net' tab selected. On the right, there are several filter and setting panels:

- Display Settings:** Options for 'Show node information' (Names, Statistics), 'Show node icon' (Business Unit, Country, None, Picture), and 'Show people by rank' (Min: 100, Max: 1000, Hide Isolates, Redraw).
- Find Person:** A search bar with a 'Highlight:' dropdown.

Dynamic networks of 400,000+ IBMers:

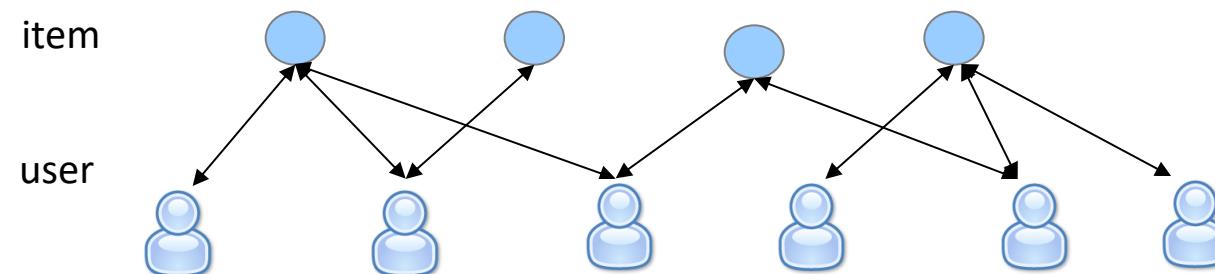
Shortest Paths
Social Capital
Bridges
Hubs
Expertise Search
Graph Search
Graph Recomm.

- On BusinessWeek four times, including being the Top Story of Week, April 2009
- Help IBM earned the 2012 Most Admired Knowledge Enterprise Award
- Wharton School study: \$7,010 gain per user per year using the tool
- In 2012, contributing about 1/3 of GBS Practitioner Portal \$228.5 million savings and benefit
- APQC (WW leader in Knowledge Practice) April 2013:

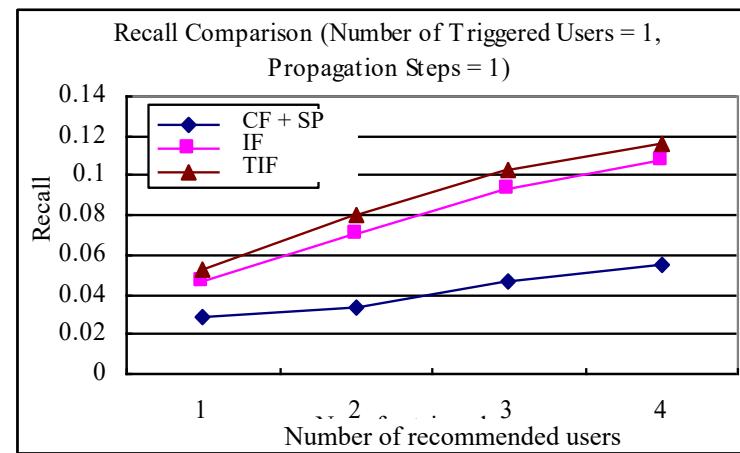
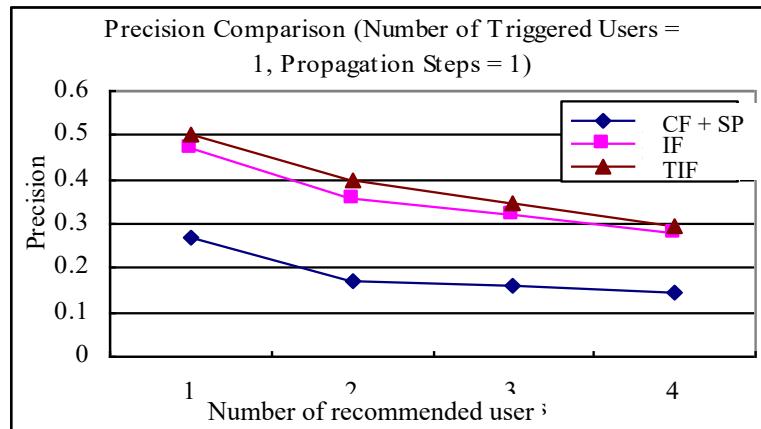
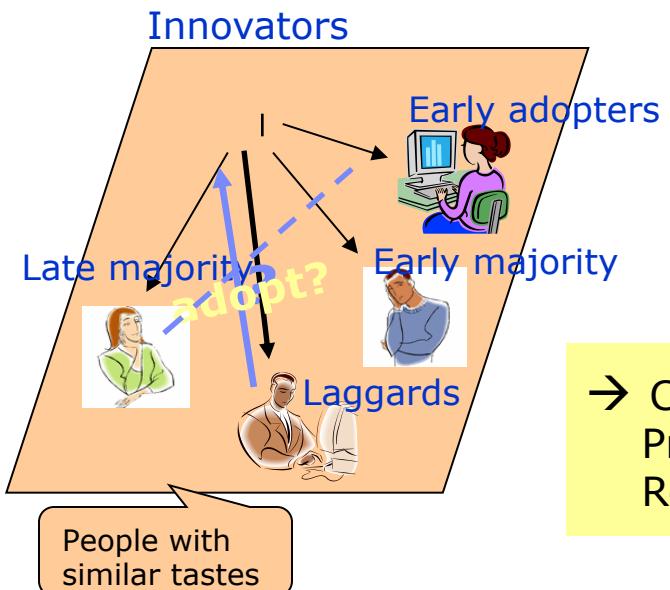
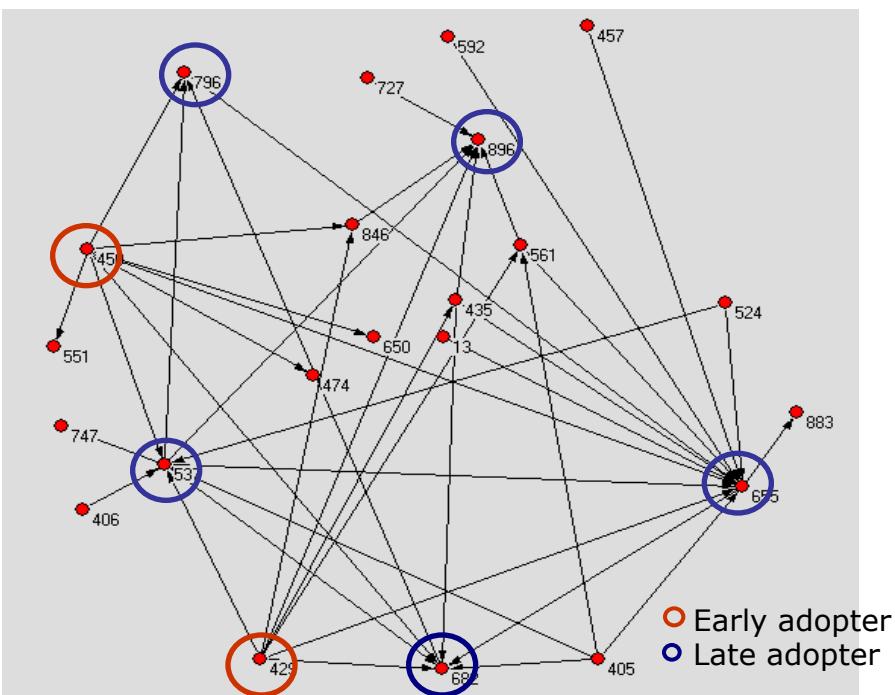
"The Industry Leader and Best Practice in Expertise Location"

Use Case 2: Recommendation

The screenshot shows the Amazon.com homepage with a user profile for "Ching Yung Lin". The top navigation bar includes links for "Your Account", "Cart", "Your Lists", "Help", and a "Gifts" button. Below the search bar, a message says "Hello, Ching Yung Lin. We have recommendations for you. (If you're not Ching Yung Lin, click here.) Make this". On the left, a sidebar titled "BROWSE" lists "Your Favorites" (Books, Software) and "Featured Stores" (Apparel & Accessories, Beauty, DVD's TV Central). The main content area features a section titled "Recommended for you" with three book recommendations: "Spikes [Reprint] Paperback by Fred Rieke", "Spiking Neuron Models Paperback by Wulfram Gerstner", and "Methods in Neuronal Modeling - 2nd Edition Hardcover by Christof Koch". Each recommendation includes a small image of the book cover and a link to "See more Recommendations".



Use Case 3: Recommendation for Commerce



Network Info Flow

Tests:
 - 1 month
 - 586 new docs
 - 1,170 users

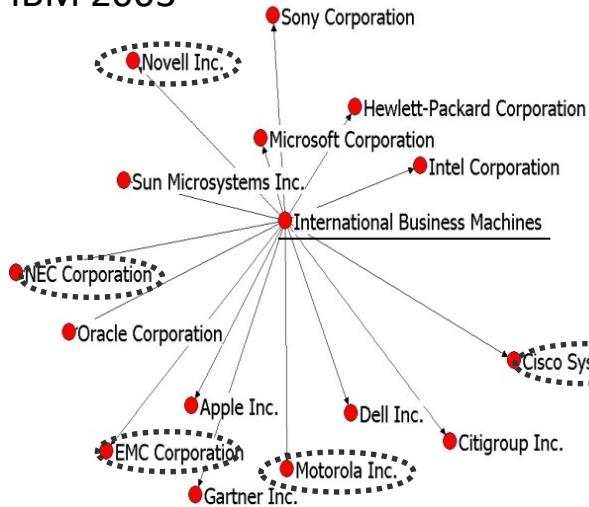
IF: Graphical Information Flow Model
 TIF: Joint Topic Detection + Information Flow Model

→ Comparing to Collaborative Filtering (CF) + Similar People
 Precision: IF is 91% better, TIF is 108% better
 Recall: IF is 87% better, TIF is 113% better

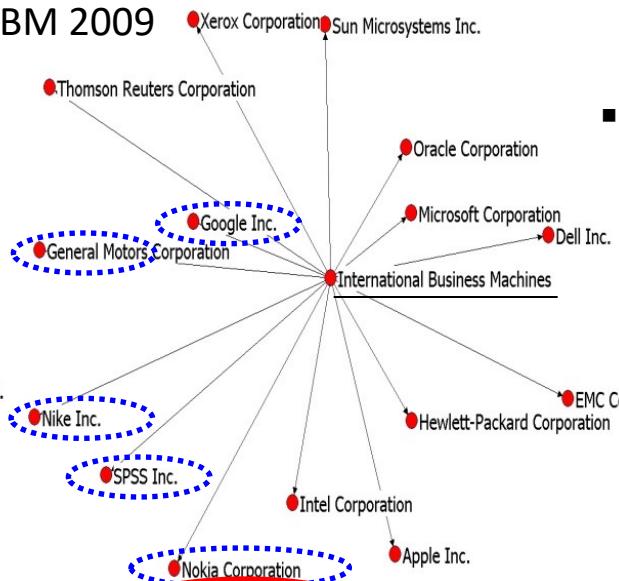
Use Case 4: Graph Analytics for Financial Analysis

Goal: Injecting Network Graph Effects for Financial Analysis. Estimating company performance considering correlated companies, network properties and evolutions, causal parameter analysis, etc.

- IBM 2003



- IBM 2009



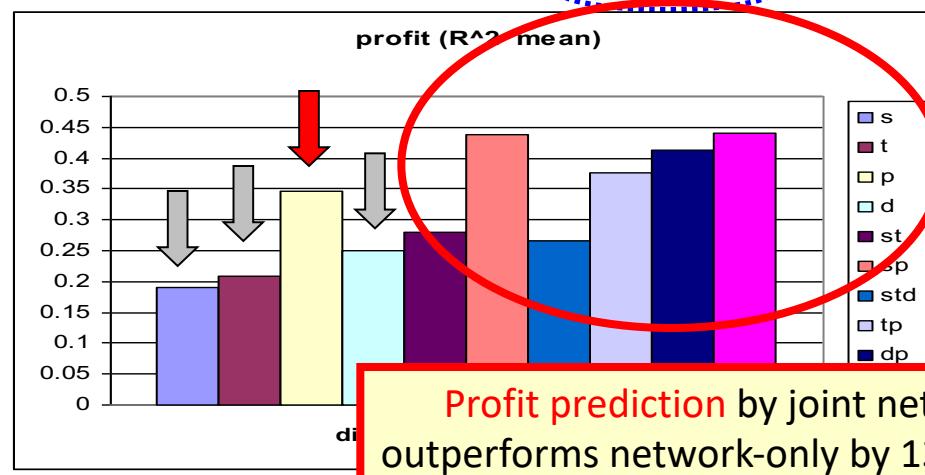
- Data Source:

- Relationships among 7594 companies, data mining from NYT 1981 ~ 2009

Targets: 20 Fortune companies' normalized Profits

Goal: Learn from previous 5 years, and predict next year

Model: Support Vector Regression (RBF kernel)



Network feature:

s (current year network feature),
t (temporal network feature),
d (delta value of network feature)

Financial feature:

p (historical profits and revenues)

Profit prediction by joint network and financial analysis outperforms network-only by 130% and financial-only by 33%.

Use Case 5: Social Media Monitoring

Ching-Yung Lin | Search www.ibm.com

System G SMISC Social Media Monitoring

Home | Live | Forensics

Select CIO Catetory(-ies): EXECDB BLADE HRTEANT IBM SecurityAnalysis SWG WATSON or Word: Egypt language: Arabic

IBM CIO monitoring categories

Monitoring filter

Total Tweets: 231
Positive: 35 15%
Negative: 31 13%

EGYPT wearing @RawyaRageh beauty brutality More || Am Egypt's 12 police hijab Egypt than Cairo dozen sponge allege Port Egypt than Cairo you my Egyptian Said egypt lady call

Saloom Butilla @SaloomButilla
إثناء المقربين الفرقة في RT @Lion_King_Bhr: في 19/2/2013 #الجرحين على المرافق العامة ورجل الأمن #Bahrain #Egypt #Syria #KSA #UAE #News h
Translation: RT *@Lion_King_Bhr*: The traitors in Bahrain Safavid attack on public utilities and security men, 2/19/2013 *LBahrain* #Egypt *LSyria* *LKSA* *LUAE* *LNews* h *...* --Wed Feb 20 17:57:58 2013

Zenza Raggi fan-club @Zenzaclub
Private Gold 64: Cleopatra 2 // A sect that worships ancient Egypt is attempting to bring Cleopatra back to life... http://t.co/TcvMDiwb --Wed Feb 20 17:57:53 2013

SH_QalamSara @SH_QalamSara
مترقبة هنا RT @HebaFaroog: An #Egypt-ian beauty :) v http://t.co/59Bzb5f3 --Wed Feb 20 17:57:53 2013

Mona Metwally @monametwally
مرتضى محتاج متبر عن دم مستكتي الجامعه بالاسمايليه فضله دم اب مرجوب RT @EgyBloodBank: AB+ 01024705247 #Egypt # مصر http://t.co/5o06mtZ5.
Translation: . RT *@EgyBloodBank*: A

Live Tweets, Sentiment, Keywords

Dynamic Graphs

Zooming / Panning

Real-Time Translation, Locations, Top Retweets

GO

Research Projects | People | News |

Real-Time Translation, Locations, Top Retweets

GO

Research Projects | People | News |

IBM CIO monitoring categories

Monitoring filter

Total Tweets: 231
Positive: 35 15%
Negative: 31 13%

EGYPT wearing @RawyaRageh beauty brutality More || Am Egypt's 12 police hijab Egypt than Cairo dozen sponge allege Port Egypt than Cairo you my Egyptian Said egypt lady call

Saloom Butilla @SaloomButilla
إثناء المقربين الفرقة في RT @Lion_King_Bhr: في 19/2/2013 #الجرحين على المرافق العامة ورجل الأمن #Bahrain #Egypt #Syria #KSA #UAE #News h
Translation: RT *@Lion_King_Bhr*: The traitors in Bahrain Safavid attack on public utilities and security men, 2/19/2013 *LBahrain* #Egypt *LSyria* *LKSA* *LUAE* *LNews* h *...* --Wed Feb 20 17:57:58 2013

Zenza Raggi fan-club @Zenzaclub
Private Gold 64: Cleopatra 2 // A sect that worships ancient Egypt is attempting to bring Cleopatra back to life... http://t.co/TcvMDiwb --Wed Feb 20 17:57:53 2013

SH_QalamSara @SH_QalamSara
مترقبة هنا RT @HebaFaroog: An #Egypt-ian beauty :) v http://t.co/59Bzb5f3 --Wed Feb 20 17:57:53 2013

Mona Metwally @monametwally
مرتضى محتاج متبر عن دم مستكتي الجامعه بالاسمايليه فضله دم اب مرجوب RT @EgyBloodBank: AB+ 01024705247 #Egypt # مصر http://t.co/5o06mtZ5.
Translation: . RT *@EgyBloodBank*: A

Real-Time Translation, Locations, Top Retweets

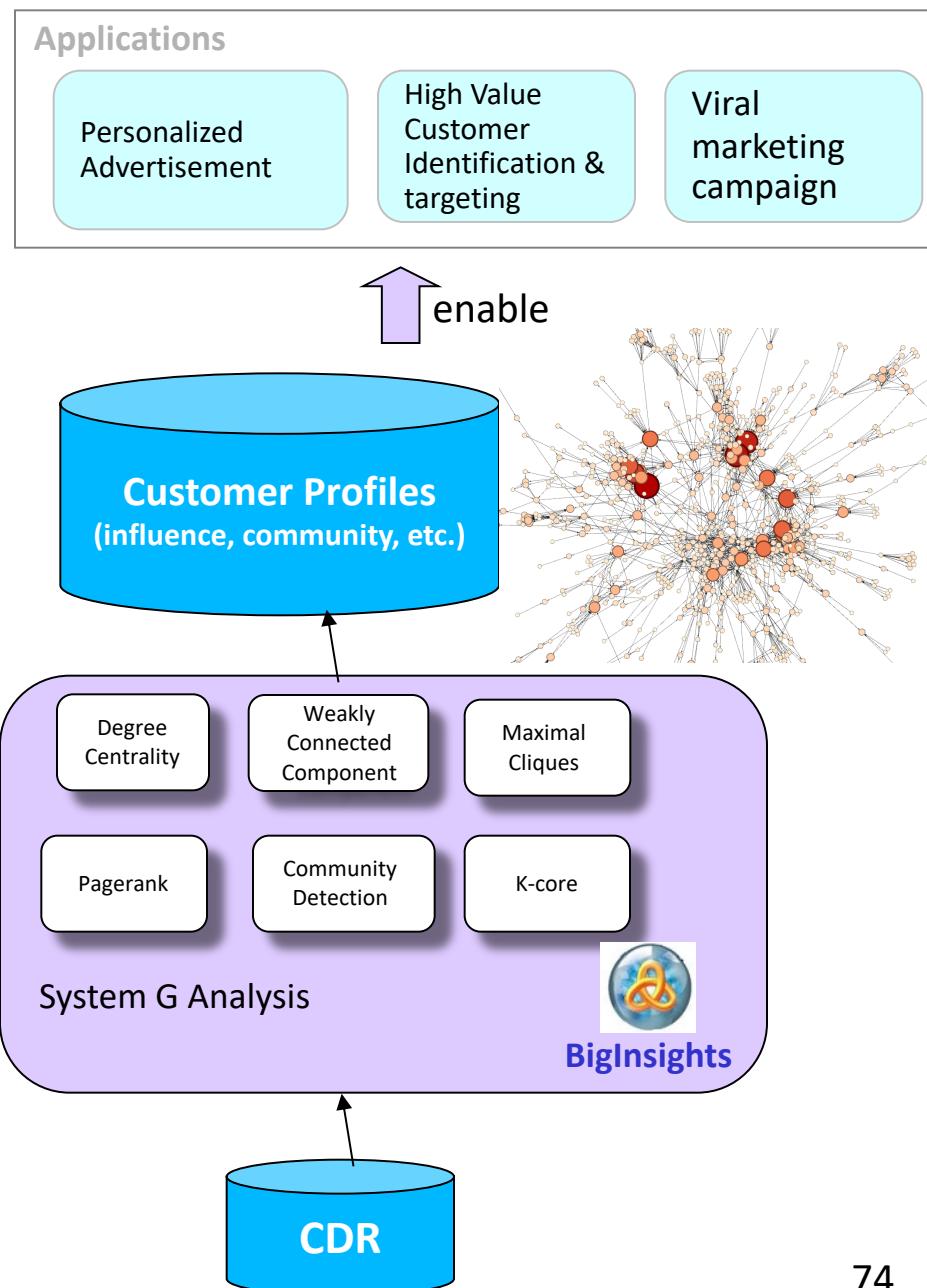
GO

Research Projects | People | News |

Use Case 6: Customer Social Analysis for Telco

Goal: Extract customer social network behaviors to enable Call Detail Records (CDRs) data monetization for Telco.

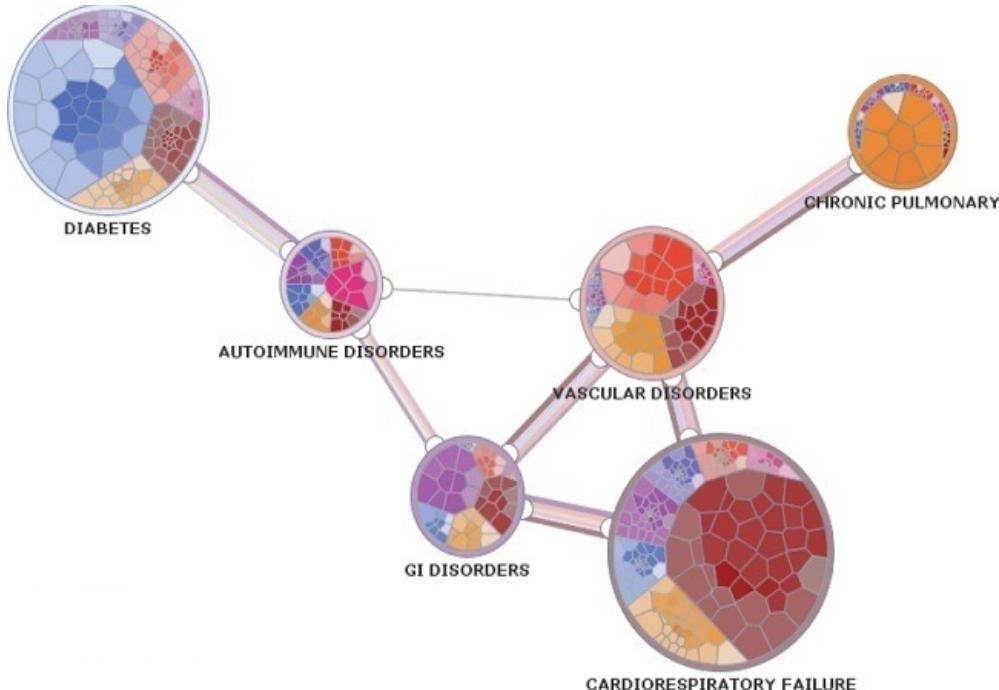
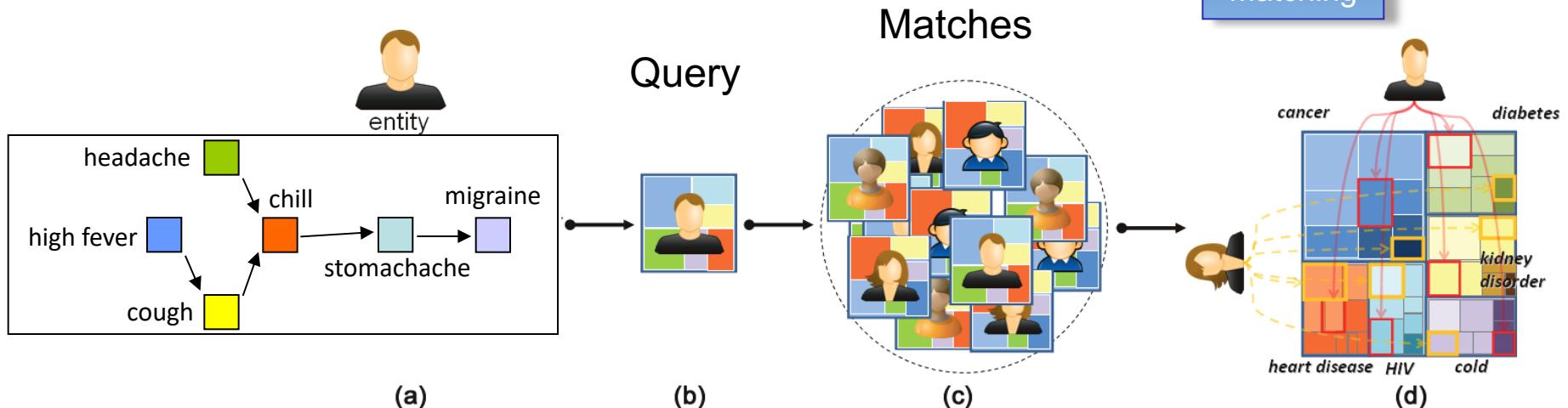
- Applications based on the extracted social profiles
 - Personalized advertisement (beyond the scope of traditional campaign in Telco)
 - High value customer identification and targeting
 - Viral marketing campaign
- Approach
 - Construct social graphs from CDRs based on {caller, callee, call time, call duration}
 - Extract customer social features (e.g., influence, communities, etc.) from the constructed social graph as customer social profiles
 - Build analytics applications (e.g., personalized advertisement) based on the extracted customer social profiles



Use Case 7: Graph Analytics and Visualization for Watson



Graph
Matching



Graph
Communities

User Case 8: Visualization for Navigation and Exploration

The screenshot shows the System G web interface at systemg.ibm.com. The top navigation bar includes Home, Overview, Analytics, Database, Middleware, Visualization, and Solutions. Below the navigation bar are three visualizations:

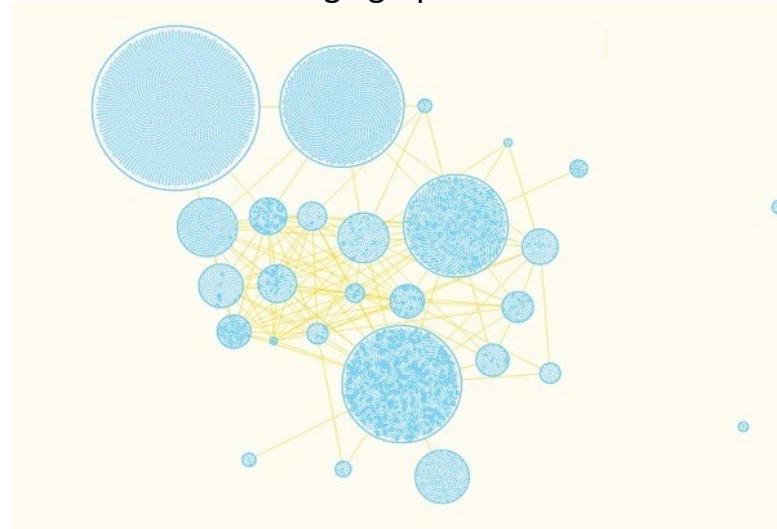
- A circular sunburst-like graph with a pink and yellow color scheme.
- A cluster-based graph where nodes are blue circles connected by yellow lines, forming several clusters.
- A network graph with nodes labeled with medical terms like "hypertension", "cigarette smoking", "obesity", "diabetes", "depression", "anxiety", and "stroke".

On the left, there's a "Recent News" sidebar with the following items:

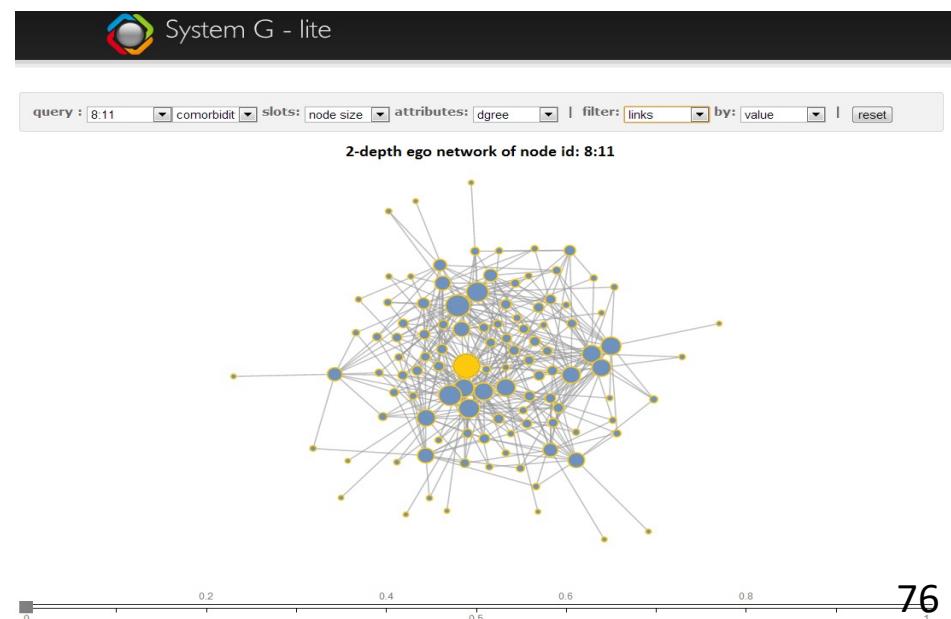
- Publications**: 8 MAY - About 100 System G papers have been uploaded to the website ...
- More Visualization Demos**: 18 MAY - More graph visualization demos have been uploaded to the SystemG website ...
- Release of systemG website**: 17 APRIL - The systemG website is online now.
- Release of Huge Graph Visualization**: 15 APRIL - The first version of BubbleNet, a huge graph visualization, is released.

The main content area includes a "Welcome to System G!" message and a "Use Cases and Solutions" section.

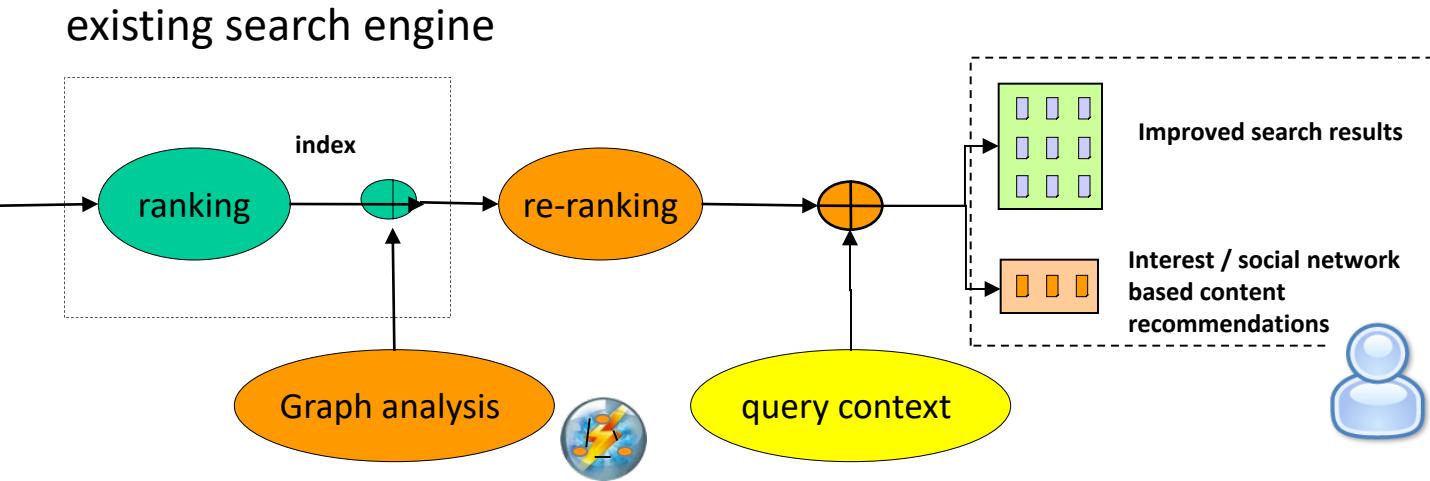
Cluster based huge graph visualization



Query based huge graph visualization



Use Case 9: Graph Search



Practitioner Portal

Translate this page: English

< Return to starting page

Refine Results

- By Tag
 - Select a tag to filter search results ⓘ
 - View as: cloud | list
- more — less

2012 analyst_report analytics bao baseline csp deliverable europe forrester ftcool gartner gbs gmu government kh leader_priority na proposal public_sector retail sales sales_tools sandt social social_business telecommunications

By Category

- Select a category to filter search results ⓘ
- Expand all | Collapse all
- Asset Type
- Audience
- Business Topics
- Client Value Method (CVM)
- Geography
- IBM Business Unit
- Industry
- Language

Search criteria

Search keywords: **social business** ⓘ

All results Social network results ⓘ

18,577 results found

1 to 25 shown

1 2 3 4 5 6 7 8 9 10 ...

Title	Relevance	Modified	Bookmarks
IBM Social Business Adoption QuickStart (U.S. English) - Proposal Insert [in Proposal and Presentation Accelerator (PPX)] ⓘ	100 %	29 Aug 2012	0
Drive the successful launch and adoption of social business software throughout your organization with a structured engagement comprised of assessments, planning and design consultation, onsite workshops, and team- and skills-building activities.			
Sales Support Information(SSI) DAGE@stibo.com			

Use Case 10: Anomaly Detection at Multiple Scales

Based on President Executive Order 13587

Goal: System for Detecting and Predicting Abnormal Behaviors in Organization, through large-scale social network & cognitive analytics and data mining, to decrease insider threats such as espionage, sabotage, colleague-shooting, suicide, etc.



THE WALL STREET JOURNAL.

Many Past Espionage Cases Had Links to

Since 2009, Justice Department lawyers have pursued at least 19 cases of corporate espionage. Most had connections to China, according to case summaries presented in a new U.S. strategy report on trade secrets Wednesday.

Among high-profile cases:

◆ In August 2012, Huiqin Jin, a former Motorola engineer, was sentenced in Illinois to four years in prison for stealing Motorola's proprietary IDEN technology for herself and for a company that works for the Chinese military.

◆ In March 2012, former DuPont scientist Tze Chao pleaded guilty in California to conspiracy to commit economic espionage, admitting he



"Enterprise Information Leakage Impacted economy and jobs" Feb 2013

npr

news > business

To Catch Worker Misconduct, Companies Hire Corporate Detectives

by AILSA CHANG

January 10, 2013 6:25 PM

"What's emerged is a multibillion dollar detective industry"
npr Jan 10, 2013

Emails

Instant Messaging

Web Access

Executed Processes

Printing

Copying

Log On/Off



Infrastructure + ~ 70 Analytics

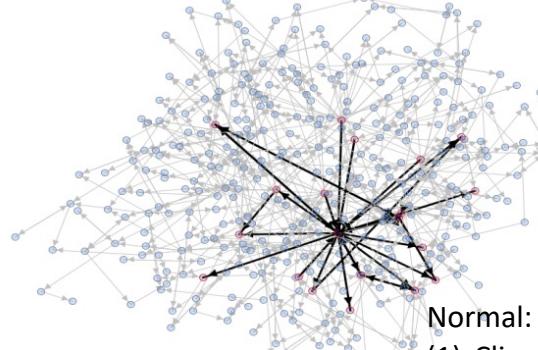
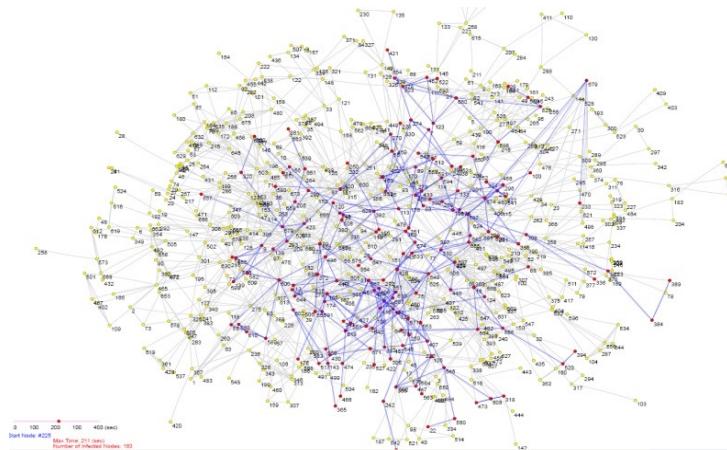
Use Case 11: Fraud Detection for Bank

Network
Info Flow

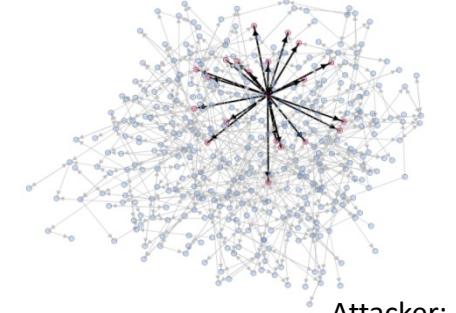
Ego Net
Features



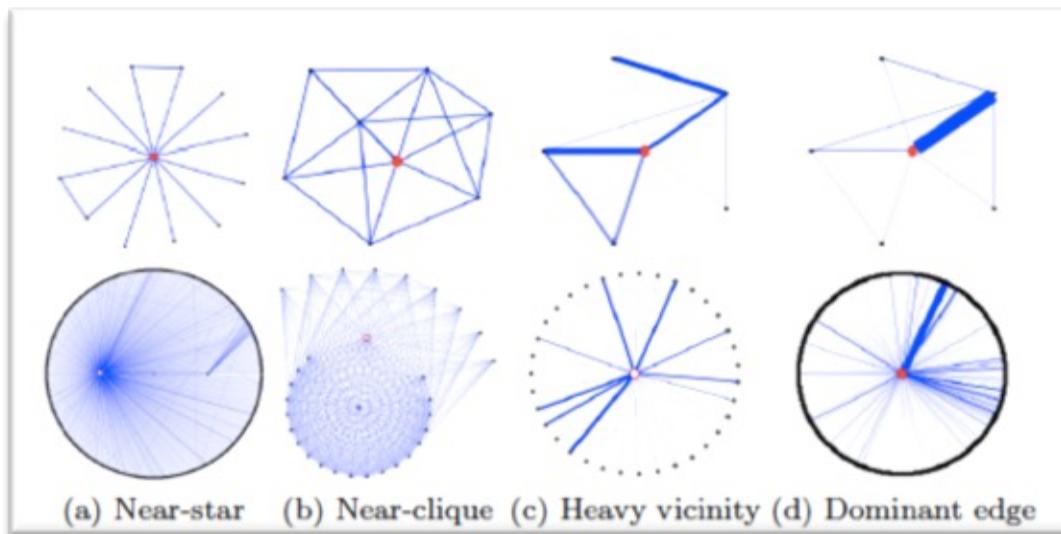
Ponzi scheme Detection



Normal:
(1) Clique-like
(2) Two-way links



Attacker:
Near-Star



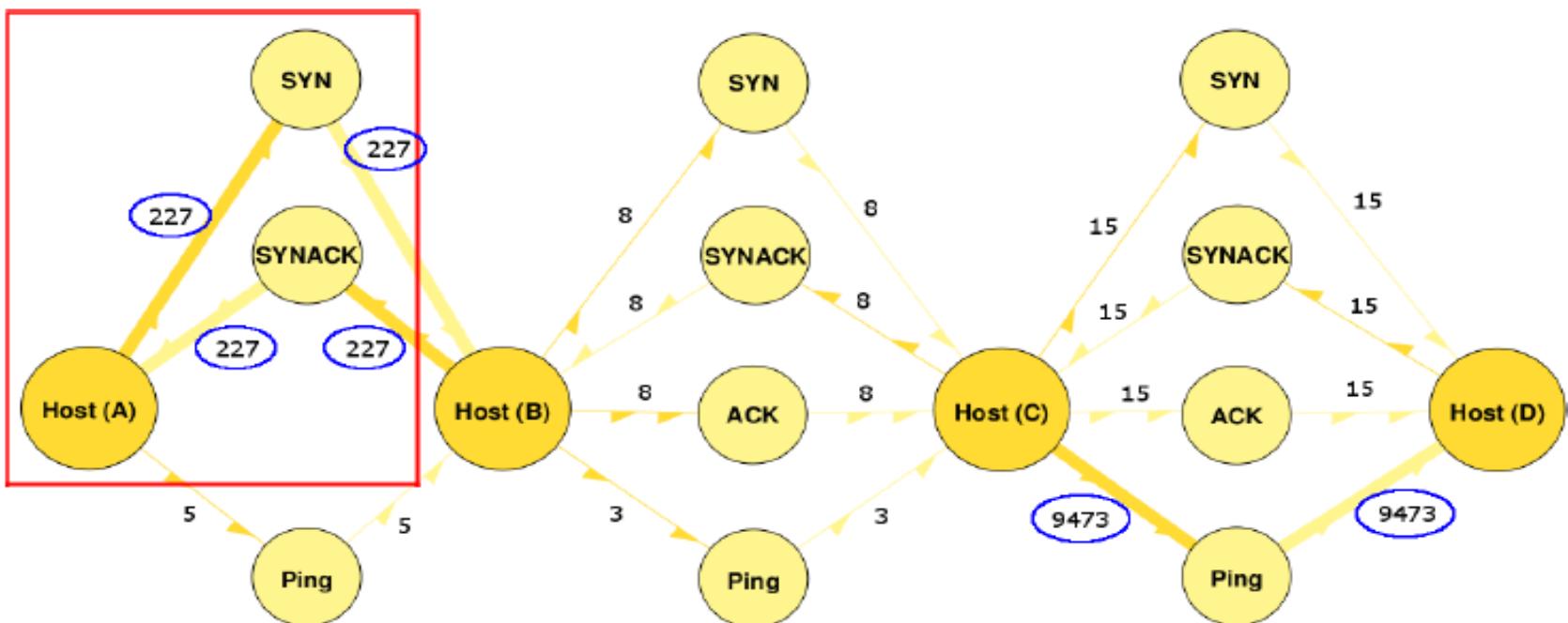
Use Case 12: Detecting Cyber Attacks

Network
Info Flow

Ego Net
Features



Detecting DoS attack

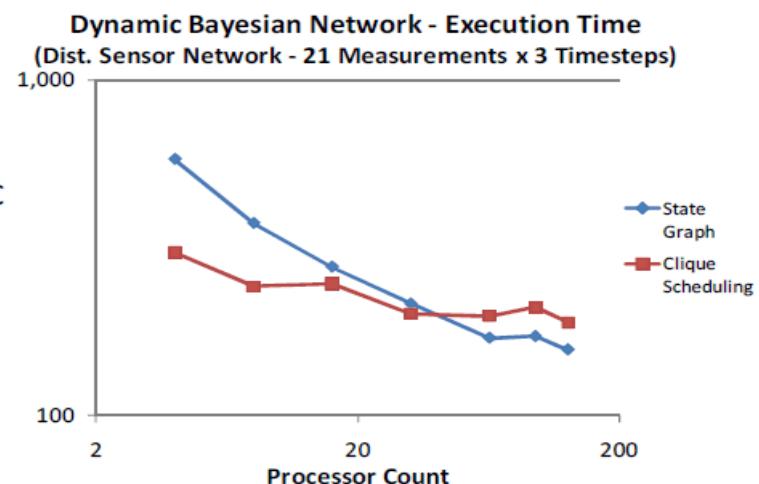
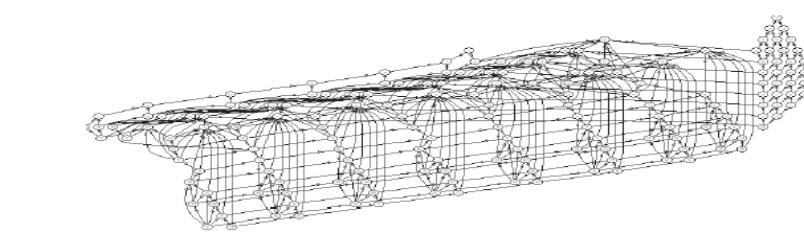
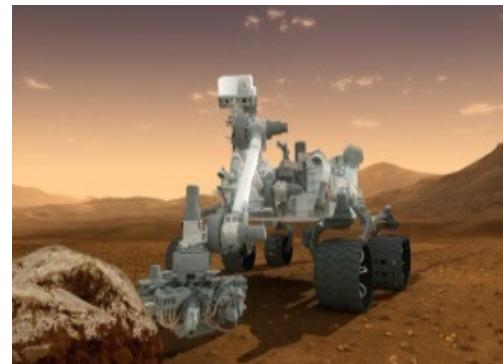


(a) Single large graph representing TCP SYN and ICMP PING network traffic, with two Denial of Service (DoS) attacks taking place.

Use Case 13: Smarter *another* Planet

Goal: Atmospheric Radiation Measurement (ARM) climate research facility provides *24x7 continuous field observations* of cloud, aerosol and radiative processes. **Graphical models** can automate the validation with improvement efficiency and performance.

Approach: BN is built to represent the dependence among sensors and replicated across timesteps. BN parameters are learned from over *15 years* of ARM climate data to support distributed climate sensor validation. Inference validates sensors in the connected instruments.



Bayesian Network

- * 3 timesteps
- * 63 variables
- * 3.9 avg states
- * 4.0 avg indegree
- * 16,858 CPT entries

Junction Tree

- * 67 cliques
- * 873,064 PT entries in cliques

Use Case 14: Cellular Network Analytics in Telco Operation

Goal: Efficiently and uniquely identify *internal* state of Cellular/Telco networks (e.g., performance and load of network elements/links) using probes between monitors placed at selected network elements & endhosts

- Applied Graph Analytics to telco network analytics based on CDRs (call detail records): **estimate traffic load** on CSP network with low monitoring overhead

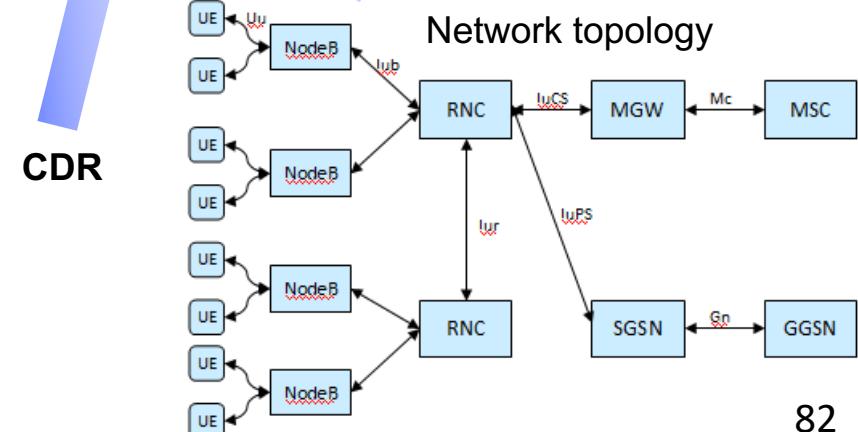
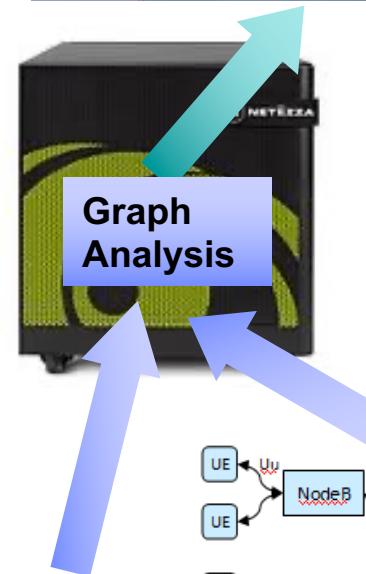
- CDRs, already collected for billing purposes, contain information about voice/data calls
- Traditional NMS* and EMS** typically lack of end-to-end visibility and topology across vendors
- Employ graph algorithms to analyze network elements which are not reported by the usage data from CDR information

Approach

- Cellular network comprises a hierarchy of network elements
- Map CDR onto network topology and infer load on each network element using graph analysis
- Estimate network load and localize potential problems



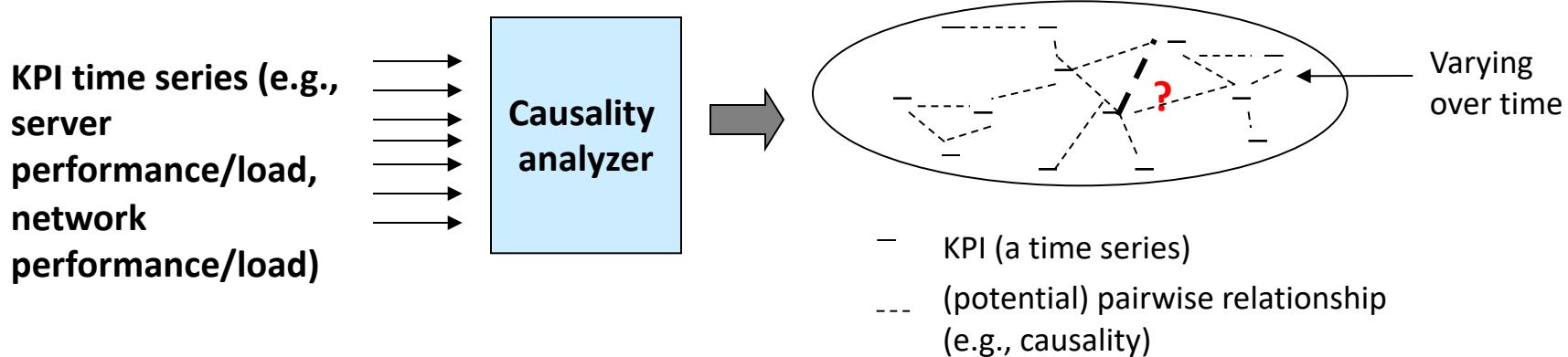
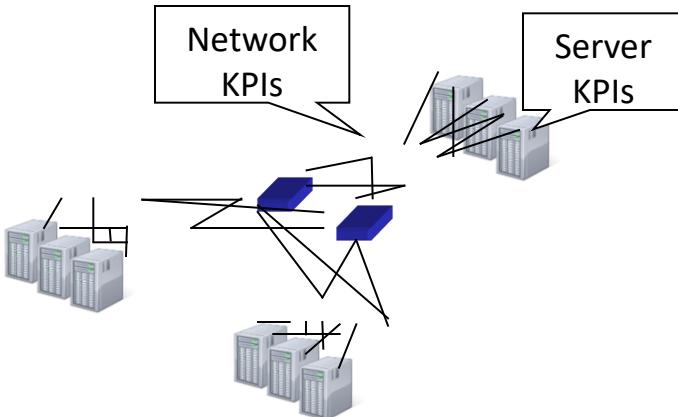
Network load level report



Use Case 15: Monitoring Large Cloud

Goal: Monitoring technology that can track the time-varying state (e.g., causality relationships between KPIs) of a large Cloud when the processing power of monitoring system cannot keep up with the scale of the system & the rate of change

- *Causality relationships (e.g., Granger causality) are crucial in performance monitoring & root cause analysis*
- *Challenge: easy to test pairwise relationship, but hard to test multi-variate relationship (e.g., a large number of KPIs)*

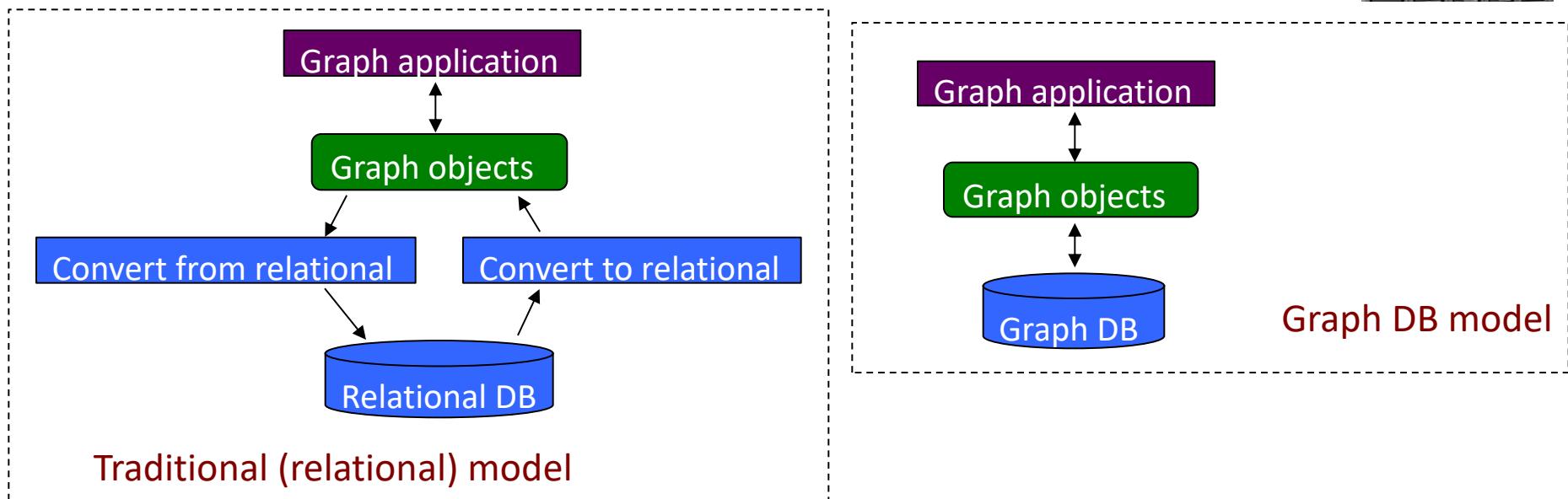


Our approach:
Probabilistic monitoring via sampling & estimation



*Select KPI pairs (sampling) → Test link existence → Estimate unsampled links based on history
→ Overall graph*

Use Case 16: Code Life Cycle Improvement



- Advantages of working directly with graph DB for graph applications
 - Smaller and simpler code
 - Flexible schema → easy schema evolution
 - Code is easier and faster to write, debug and manage
 - Code and Data is easier to transfer and maintain

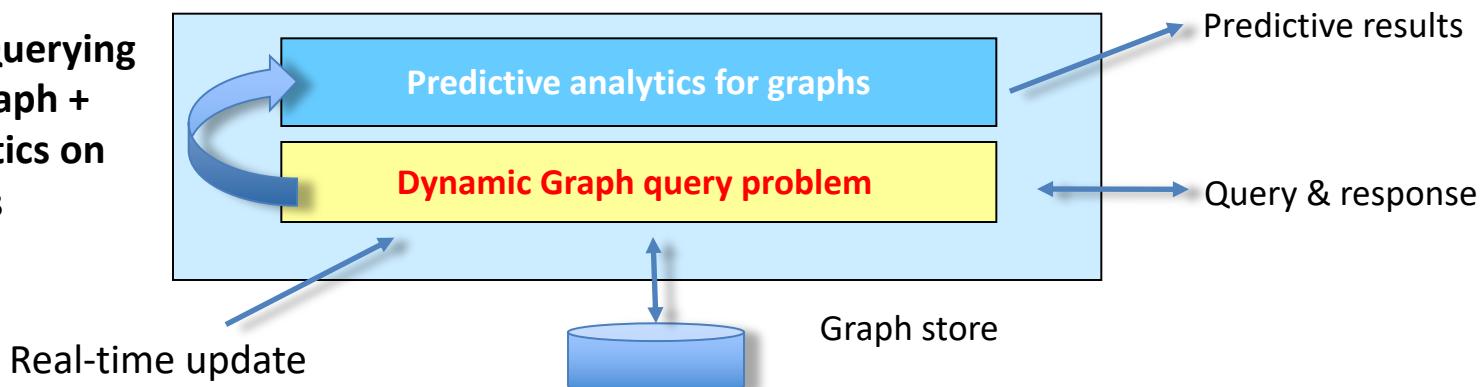
Use Case 17: Smart Navigation Utilizing Real-time Road Information

Goal: Enable unprecedented level of accuracy in **traffic scheduling** (for a fleet of transportation vehicles) and navigation of individual cars utilizing the **dynamic real-time information** of changing road condition and predictive analysis on the data

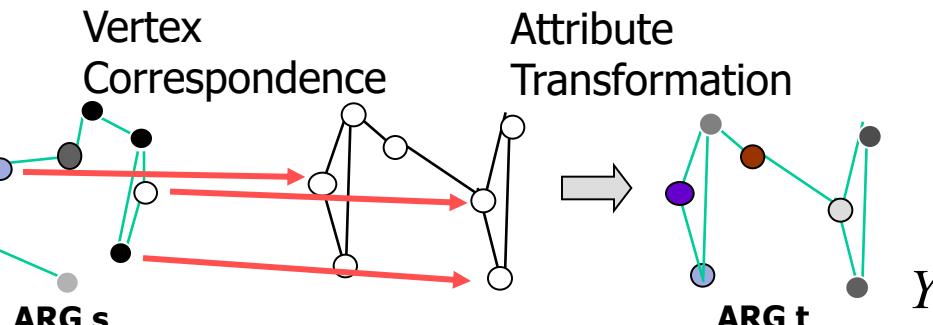
- Dynamic graph algorithms implemented in System G provide **highly efficient graph query computation** (e.g. shortest path computation) on time-varying graphs (order of magnitudes improvement over existing solutions)
- High-throughput **real-time predictive analytics** on graph makes it possible to estimate the future traffic condition on the route to make sure that the decision taken now is optimal overall



Our approach: Querying over dynamic graph + predictive analytics on graph properties



Use Case 18: Graph Analysis for Image and Video Analysis



Use Case 19: Graph Matching for Genomic Medicine

- Ongoing discussions

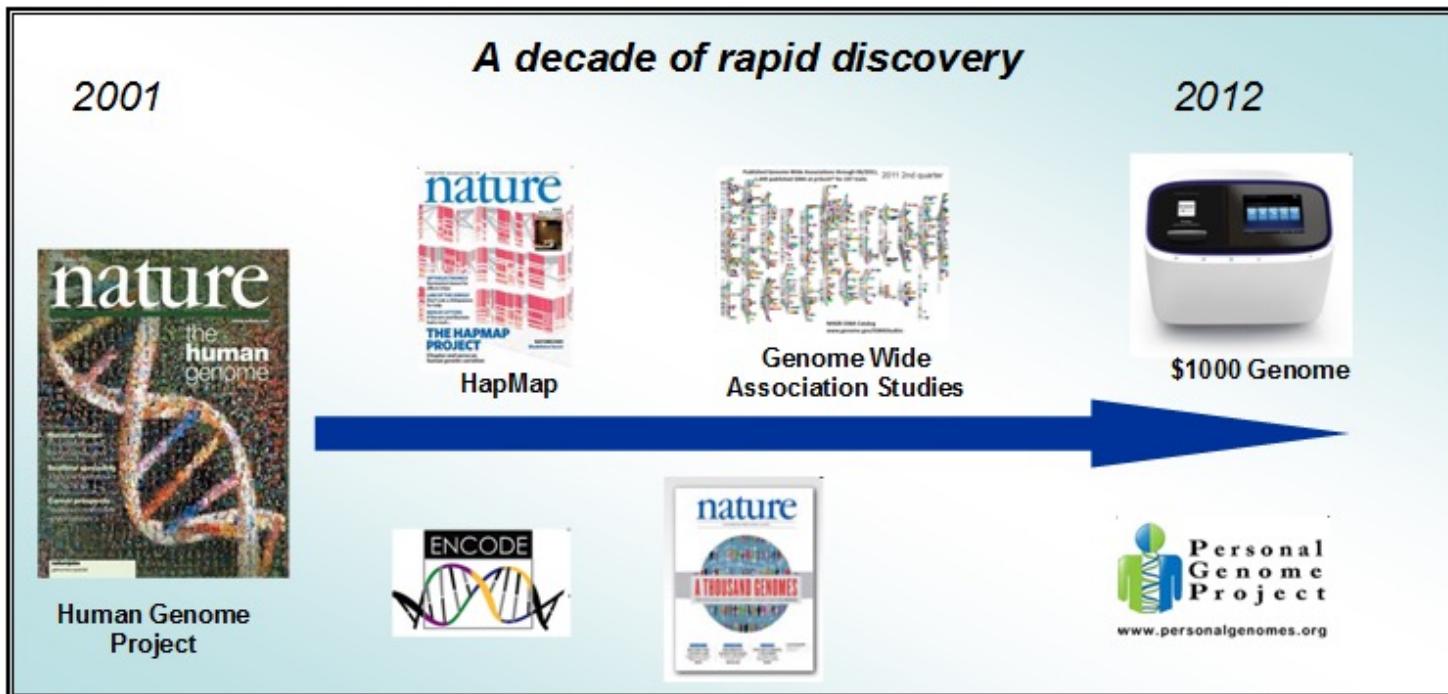
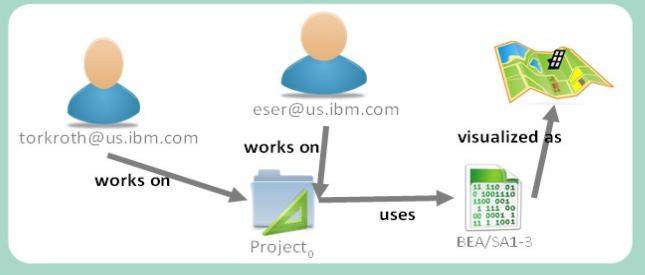


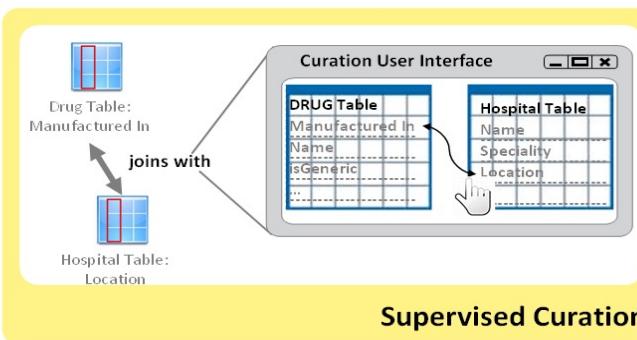
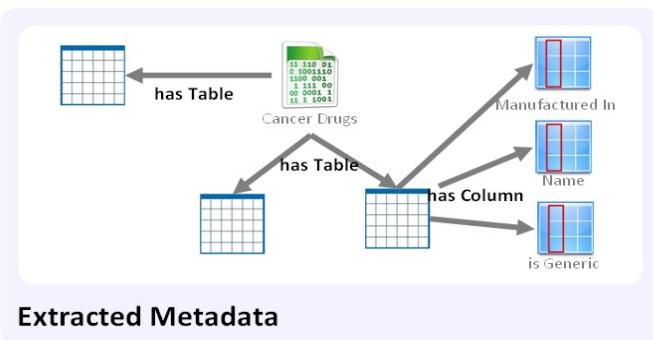
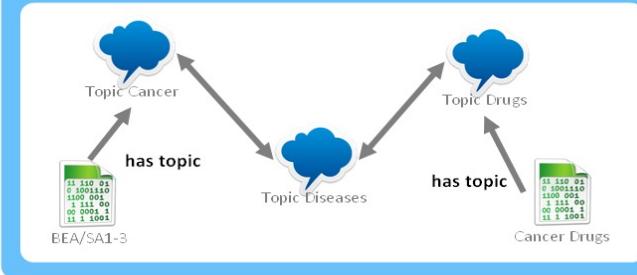
Figure 1: Since the Human Genome Project, various projects have started to reveal the mysteries of genomes and the \$1000 Genome is almost reality.

Use Case 20: Data Curation for Enterprise Data Management

Prior Collaborative Use

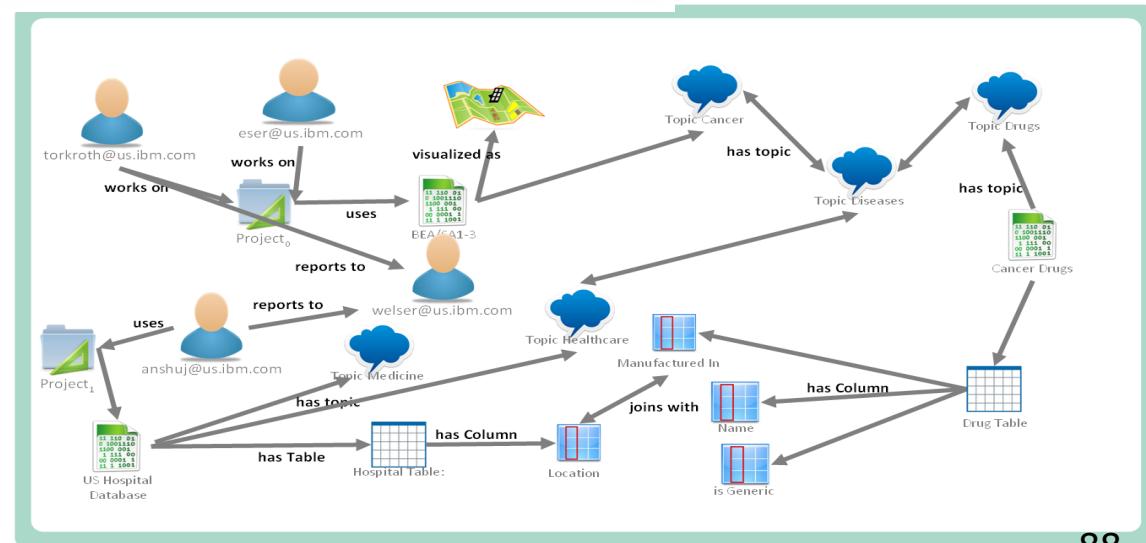
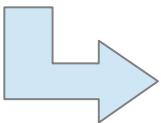


Semantic Knowledge

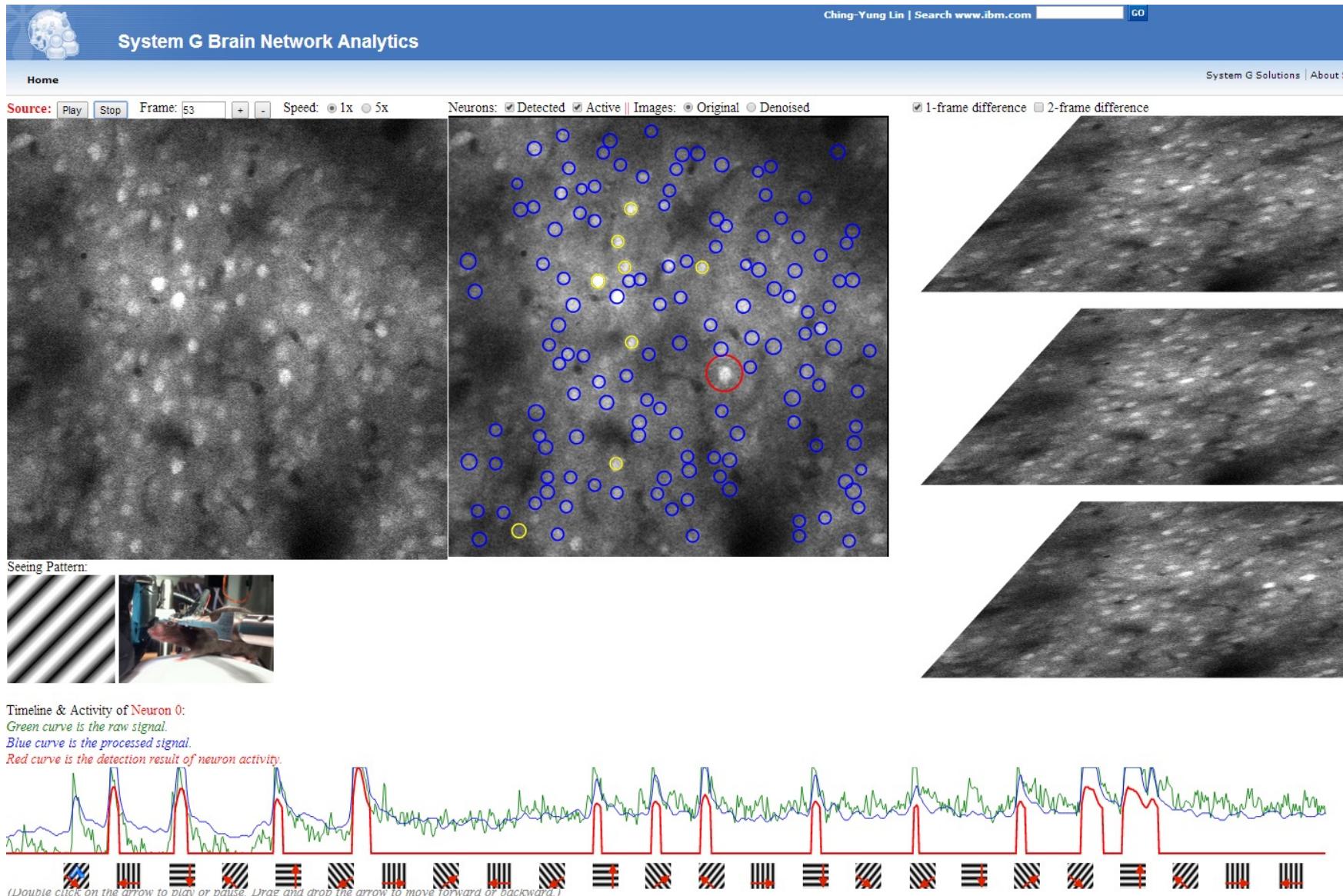


Extracted Metadata

Supervised Curation



Use Case 21: Understanding Brain Network



Use Case 22: Planet Security

- Big Data on Large-Scale Sky Monitoring

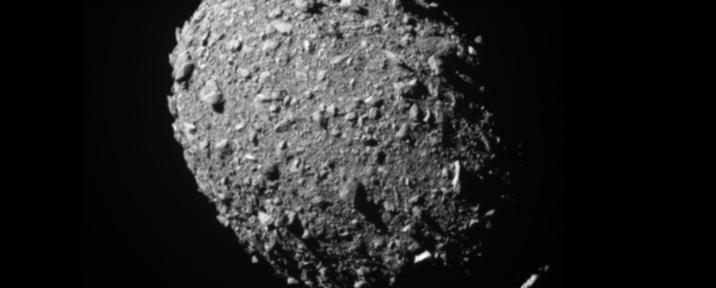


Photograph by Rob Ratkowski for the PS1SC

Dangers from space Learn about the threat to Earth from asteroids & comets and how the Pan-STARRS project is designed to help detect these NEOs. Learn more...	1,400,000,000 pixels Pan-STARRS has the world's largest digital cameras. Read about them here...	The PS1 Prototype PS1 goes operational and begins science mission PS1 Science Consortium formed... PS1SC Blog PS1 image gallery
--	---	--

NASA's DART Mission Hits Asteroid in First-Ever Planetary Defense Test

DART (Double Asteroid Redirection Test)



Sep 26, 2022
RELEASE 22-100

NASA's DART Mission Hits Asteroid in First-Ever Planetary Defense Test

[f](#) [t](#) [in](#) [p](#) [+](#)

After 10 months flying in space, NASA's Double Asteroid Redirection Test (DART) – the world's first planetary defense technology demonstration – successfully impacted its asteroid target on Monday, the agency's first attempt to move an asteroid in space.

Mission control at the Johns Hopkins Applied Physics Laboratory (APL) in Laurel, Maryland, announced the successful impact at 7:14 p.m. EDT.

As a part of NASA's overall planetary defense strategy, DART's impact with the asteroid Dimorphos demonstrates a viable mitigation technique for protecting the planet from an Earth-bound asteroid or comet, if one were discovered.

"At its core, DART represents an unprecedented success for planetary defense, but it is also a mission of unity with a real benefit for all humanity," said NASA Administrator Bill Nelson. "As NASA studies the cosmos and our home planet, we're also working to protect that home, and this international collaboration turned science fiction into science fact, demonstrating one way to protect Earth."

DART targeted the asteroid moonlet Dimorphos, a small body just 530 feet (160 meters) in diameter. It orbits a larger, 2,560-foot (780-meter) asteroid called Didymos. Neither asteroid poses a threat to Earth.

<https://www.nbcnews.com/video/nasa-s-dart-spacecraft-crashes-into-asteroid-149320773570>

Questions?