COLLABORATE14
TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

#C14LV

Easing Large-Scale Knowledge Discovery in Oracle Endeca with Big Data and Semantic Spaces

Session ID#: 14097

Prepared by:
Kenton Troy Davis
Director of Infrastructure
BIAS Corporation

Personal Information

- Kenton Troy Davis on *LinkedIn (https://www.linkedin.com/pub/kenton-davis/1/b82/40a)*
- Data Science Enthusiast
- Co-patented “Dynamic Auditing” (Pat.#7814075)
- Co-patented "Element Management System for Heterogeneous Telecommunications Network (Pat.#6260062)"

The Synopsis: Part 1

A major utility company actively uses Big Data for processing of sensor data along their power grids.

A collaborative effort between the Business Intelligence team, Data Scientists, and Marketing Analysts at the company employs a labor-intensive process to pull news feeds from the Internet via Google APIs or manual downloads. The news feeds are inspected every day for reconnaissance and knowledge discovery of what competitors are doing.

The number of news feeds is large and contains too much redundant content.

The Synopsis: Part 2

- Goal is to place a smaller memory and processing burden upon Endeca without losing relevant information.
- Proposal is to offload text mining using Map Reduce jobs before ingestion into Endeca.
- Text mining can be used to cluster multiple documents into groups. This enables analysts to easily pick just one or two documents from a group to simplify knowledge discovery (e.g. gist, sentiment).
- Pruning redundant information before visualization in Endeca facilitates more time available for targeted filtering and searching. Less documents are needed to arrive at meaningful conclusions.

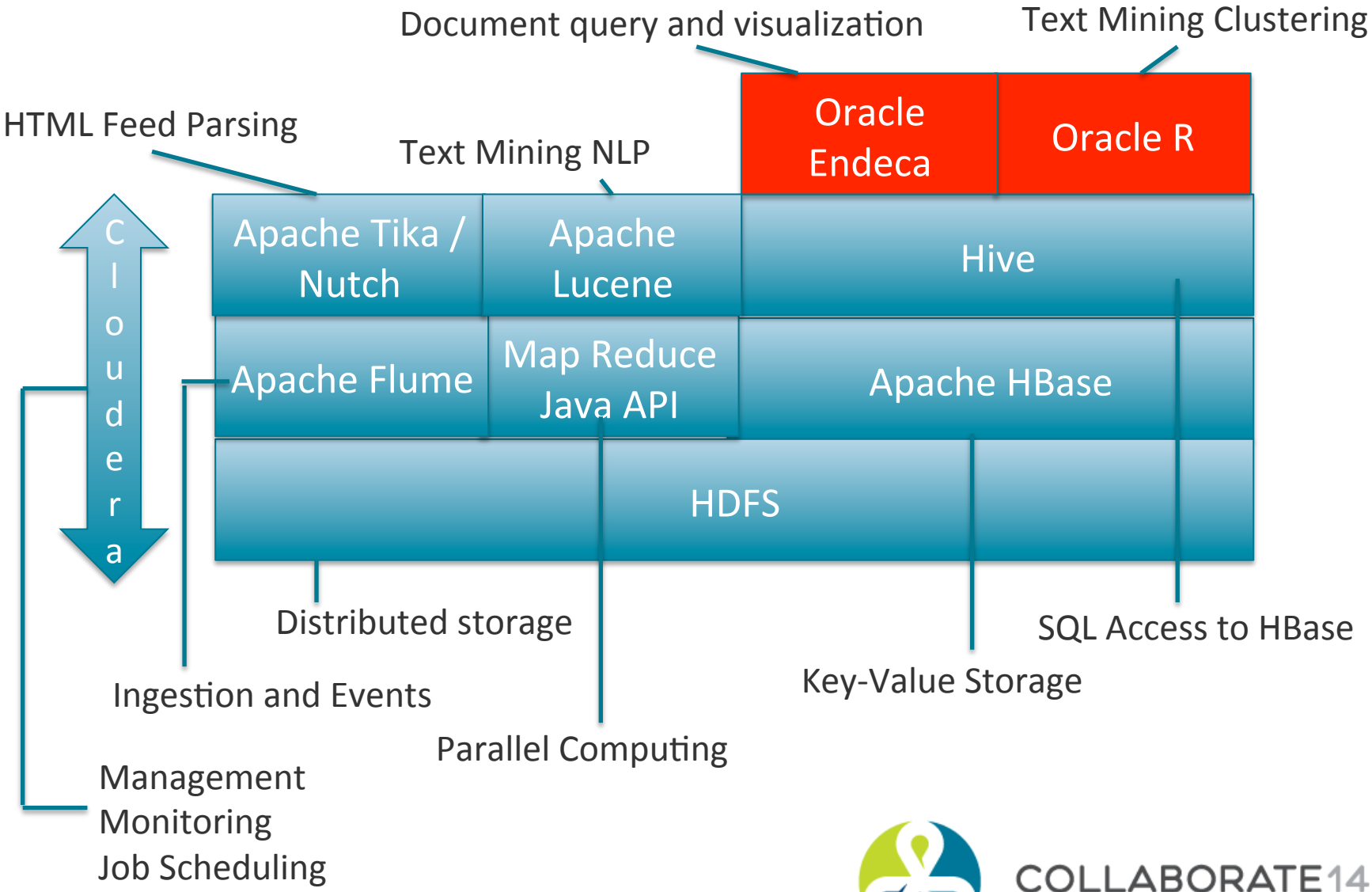
Customer Objectives and Predispositions

- Customer wants an Enterprise Data Hub in Hadoop.
- BI team likes Oracle Endeca Information Discovery, but wants to reduce the tool's in-memory processing burden by offloading some of the text mining beforehand.
- Programming team is biased towards the use of Java and Apache open source for all data staging and processing.
- In-house data scientists refuse to use Apache Mahout and insist upon the use of Oracle R.
- Customer wants an initial build using commodity hardware before considering the use of the Oracle Big Data Appliance or Oracle Exalytics.

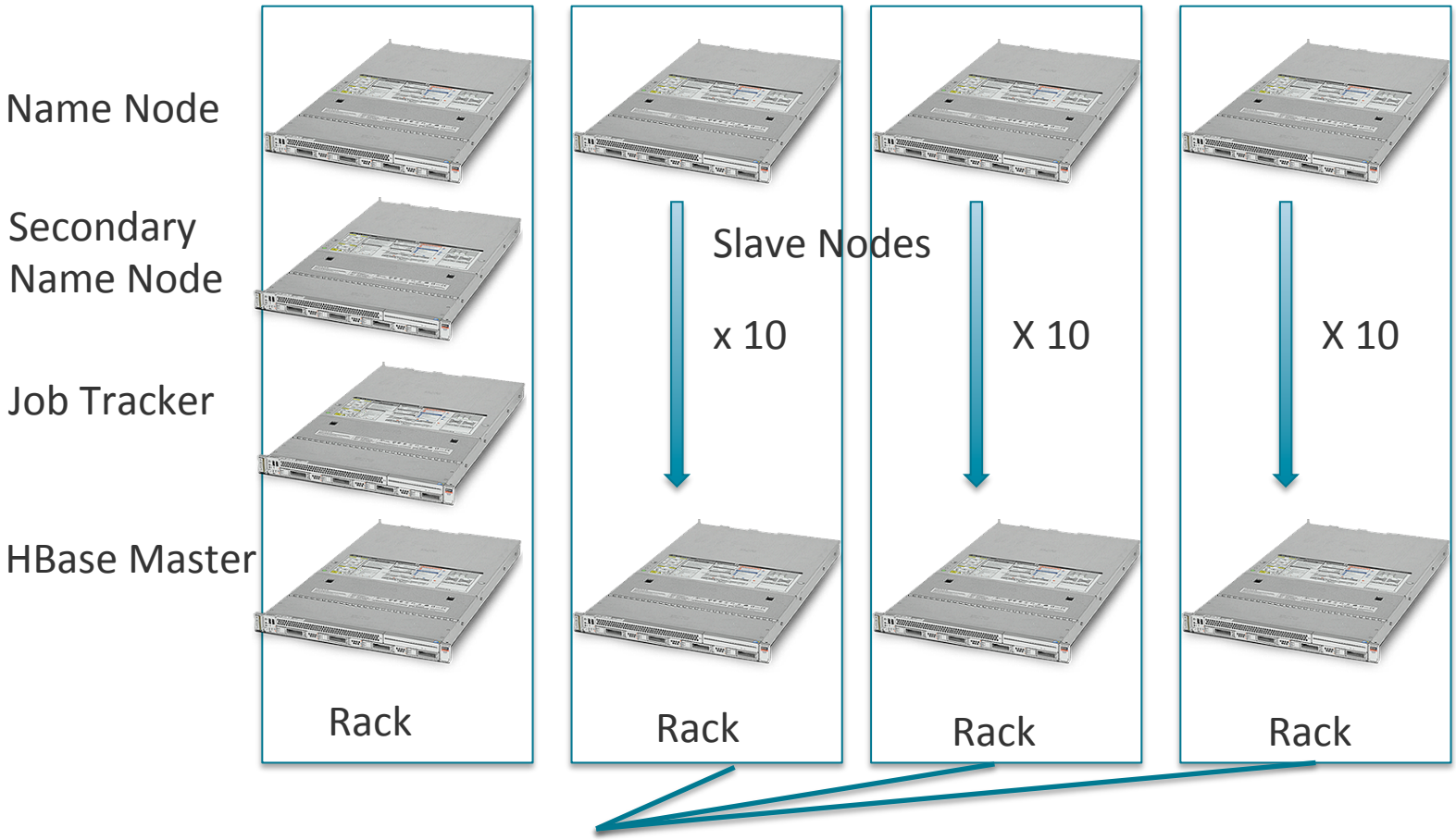
Core Components of the Solution

- The Hadoop Distributed File System (HDFS) is a scalable and portable file system optimized for data storage across a commodity cluster.
- The Map Reduce framework is a computing model for large scale data processing via parallelized job execution.
- A NoSQL Database provides data storage without the use of tabular relations used in RDBMS (e.g. key-value, graph).
- Text Mining uses lexical analysis and statistics to gather information from structured or unstructured text (e.g. news feeds, HTML, Word documents, Twitter). Useful for clustering documents and deriving themes and sentiment.

Building Blocks

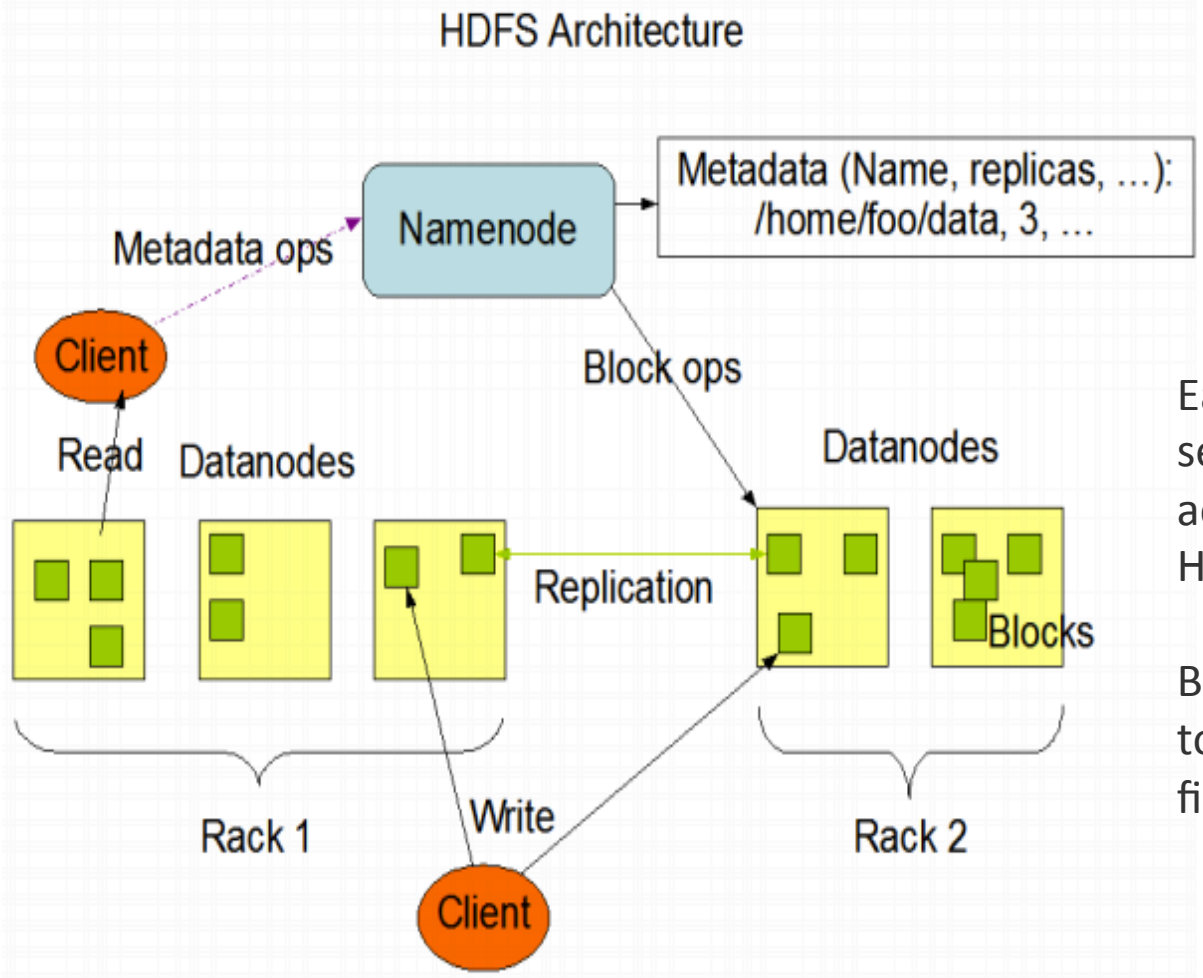


Hardware Deployment: Hadoop Cluster



Data Nodes, Task Trackers, and HBase Region Servers deployed together

Hadoop Distributed File System (HDFS)



FsImage
EditLog

Each file is stored as a sequence of blocks across Datanodes in the Hadoop cluster.

Blocks are replicated for fault tolerance based upon the file's replication factor.

Hardware Deployment: Capacity Planning

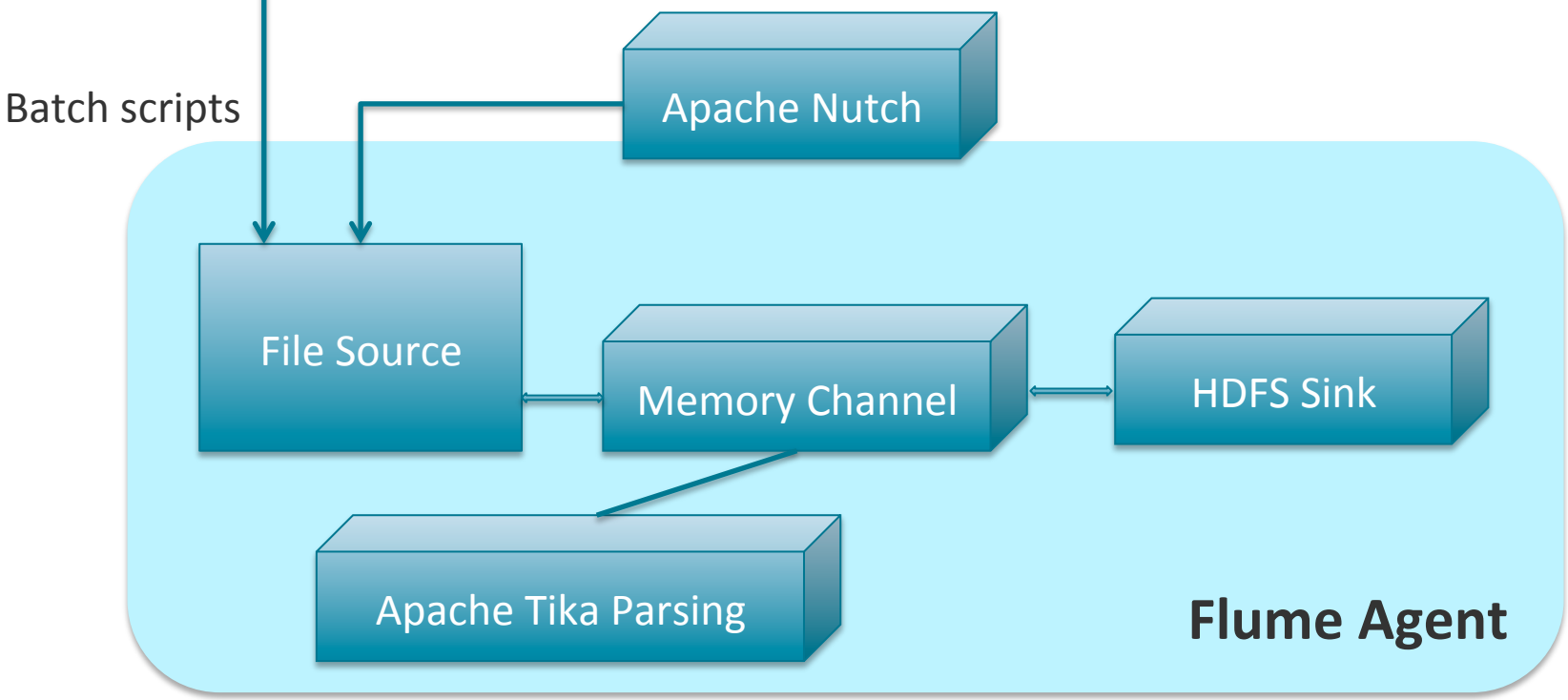
- 34 Nodes of Sun Fire X4170 M3 Servers (Oracle x86).
- Per server: four 10GbE cards, two Xeon E5-2600 CPUs, 512 GB of RAM.
- Each server connected via FCP to HA Cluster Pair of NetApp FAS3170 Controllers hosting 450TB of raw storage.
- Plan for 50GB of growth and up to 1,500 Hive queries per day.

Data Processing and Text Mining Pipeline

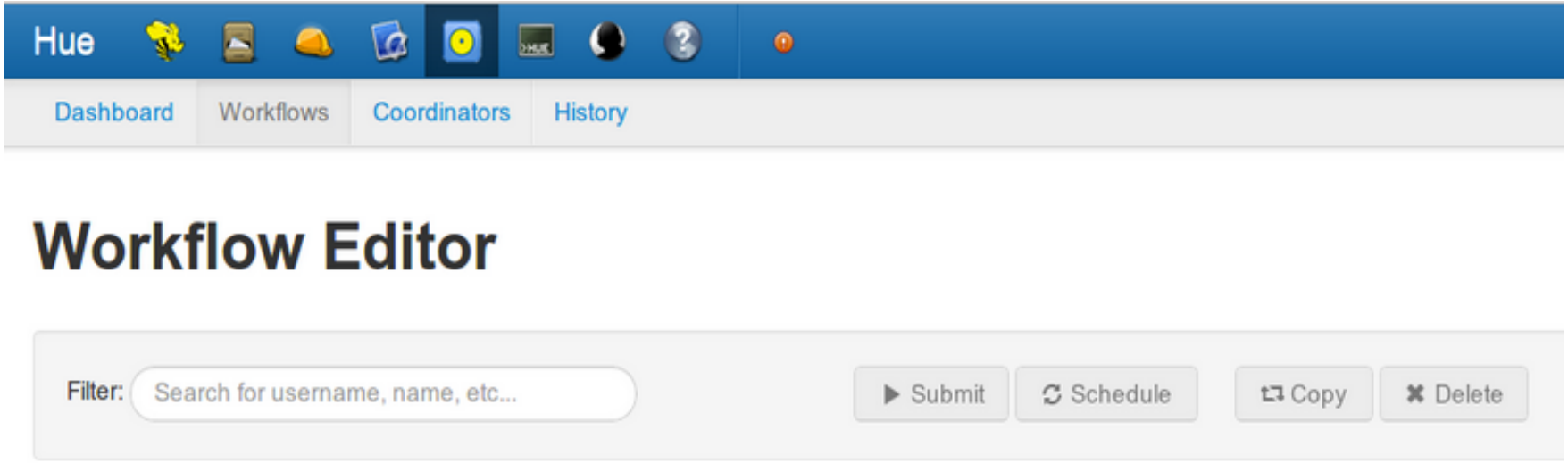
- Ingest HTML feeds with Apache Flume.
- Schedule Map Reduce Jobs.
- Process text documents with Map Reduce.
- Create a semantic space using Oracle R.
- Explore the space in Oracle Endeca Information Discovery

Ingest HTML Feeds with Apache Flume

```
SEED_DATA=`curl http://www.prnewswire.com/<some-article>.html | grep
“articleBody"`
echo ${SEED_DATA} > /var/spool/prnewswire/<some-article>_seed
```



Schedule Map Reduce Jobs

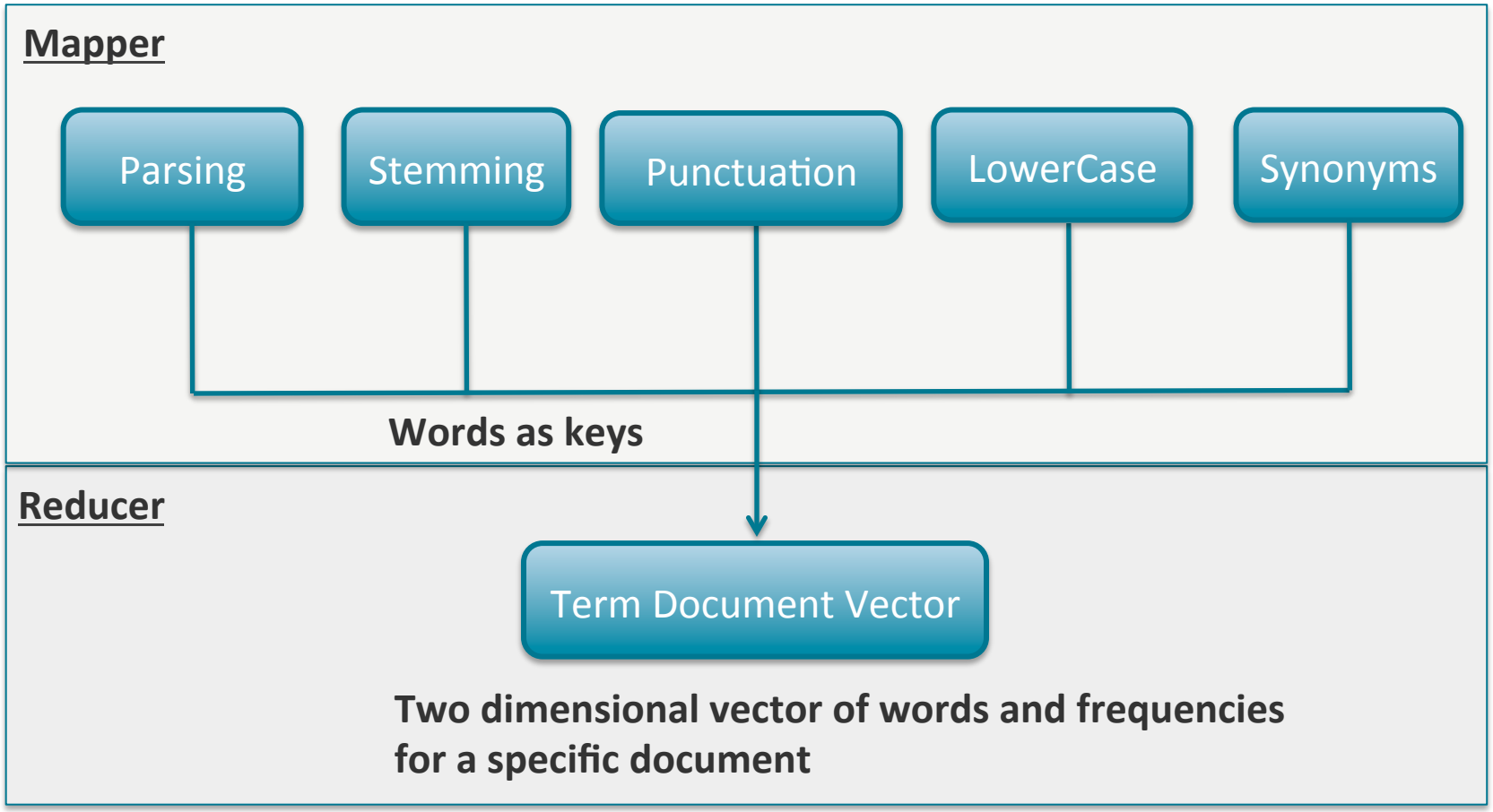


Hue is a Web application that integrates with Apache Oozie.

Use it to:

- Submit and monitor workflows and Map Reduce jobs to process the HTML text ingested.
- Browse HDFS files used for input and output.

Process Text Documents with Map Reduce



Text Processing Concepts

- Stemming is the process of reducing words to their base form (e.g. fishing, fished = fish). Leads to query expansion. Sometimes the resulting stem is not a word itself.
- Part-of-Speech (POS) Tagging marks up a text corpus with tags to identify the part of speech (e.g. noun, verb, adjective, preposition, article, pronoun, article).
- Stemming and POS Tagging are dependent upon the language used.
- Stemming and POS Tagging are the filters applied before word frequency analysis.
- The goal is to develop a semantic space which is a mathematical representation of a large body of text.

Example Mapper Code

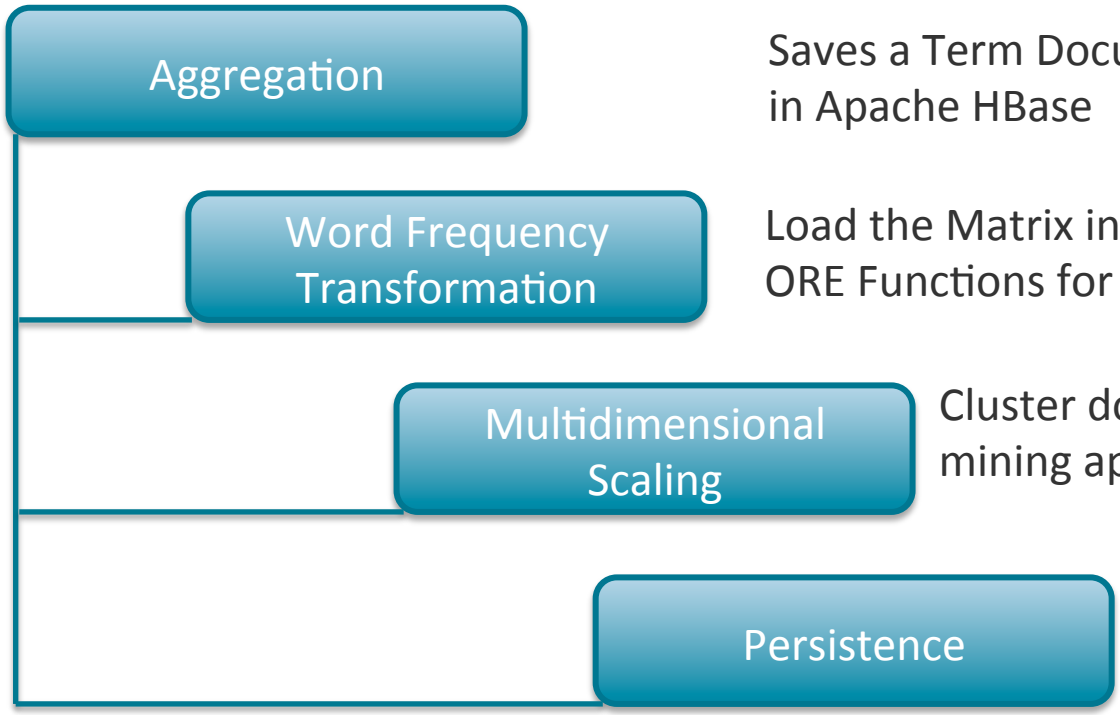
```
public class CorpusMapper extends MapReduceBase
        implements Mapper<LongWritable, Text, Text, IntWritable>
{
  public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
                  Reporter report) throws IOException
  {
    StandardAnalyzer analyzer = new StandardAnalyzer(Version.LUCENE_40);
    TokenStream tStream = analyzer.tokenStream("content", new StringReader(value.toString()));
    PorterStemFilter porter = new PorterStemFilter(tStream);
    OffsetAttribute offset = tStream.addAttribute(OffsetAttribute.class);
    CharTermAttribute charTerm = tStream.addAttribute(CharTermAttribute.class);
    tStream.reset();

    IntWritable one = new IntWritable(1);
    Text word = new Text();
    while(porter.incrementToken())
    {
      int startOffset = offset.startOffset();
      int endOffset = offset.endOffset();
      String term = charTerm.toString();
      word.set(term);
      output.collect(word, one);
    }
  }
```

Example Reducer Code

```
public class CorpusReducer extends MapReduceBase
    implements Reducer<Text, IntWritable, Text, IntWritable>
{
  public void reduce(Text key, Iterator<IntWritable> values,
                     OutputCollector<Text, IntWritable> output, Reporter report)
    throws IOException
  {
    int sum = 0;
    while (values.hasNext()) {
      sum += values.next().get();
    }
    output.collect(key, new IntWritable(sum));
  }
}
```

Create a Semantic Space



Saves a Term Document Matrix in Apache HBase

Load the Matrix into Oracle R using ORE Functions for Hive

Cluster documents using text mining approach of SVD.

Save the results from Oracle R (i.e. the semantic space) into Apache HBase.

Aggregation using Java and HBase

- Using Java, access HDFS and load each Term Document Vector that was produced as output from Map Reduce.
- Combine all of the Vectors into a Term Document Matrix and save within Apache HBase.
- A Term Document Matrix identifies all of the documents as rows and all of the words as columns. For example, the word “utility” occurs five times in Document #1 as the matrix below reveals:

	"utility"	"oil"	"car"	"consumer"	.....
Document 1	*5*	*1*	2	1	
Document 2	*3*	*0*	0	6	
Document 3	4	1	2	2	

.....

Word Frequency Transformation in Oracle R

- Access the Term Document Matrix in HBase from R using the ORE Functions for Hive.
- Some words appear more frequently than others. High frequency doesn't necessarily reflect how important the word is in distinguishing one document from another in a corpus.
- Term Frequency – Inverse Document Frequency (tf-idf) is a transformation that is applied to a corpus to decrease the weight of terms that occur very frequently and increase the weight of those that are more rare.

$$idf(i,j) = \begin{cases} 0 & \text{if } wf_{i,j} = 0 \\ (1 + \log(wf_{i,j}))\log\frac{N}{df_i} & \text{if } wf_{i,j} \geq 1 \end{cases}$$

Multidimensional Scaling in Oracle R

- Large numbers of words and documents produce a matrix of many dimensions. Data science (i.e. Latent Semantic Indexing in this case) is used to scale down the number of dimensions needed for analysis.

- Singular Value Decomposition (SVD) of the matrix reduces the number of dimensions into linearly independent combinations or partitions that explain the most variance between documents.

- The distance between each row in the resulting SVD matrix is calculated to give a numerical measure of how close each document is to the other documents in the corpus.

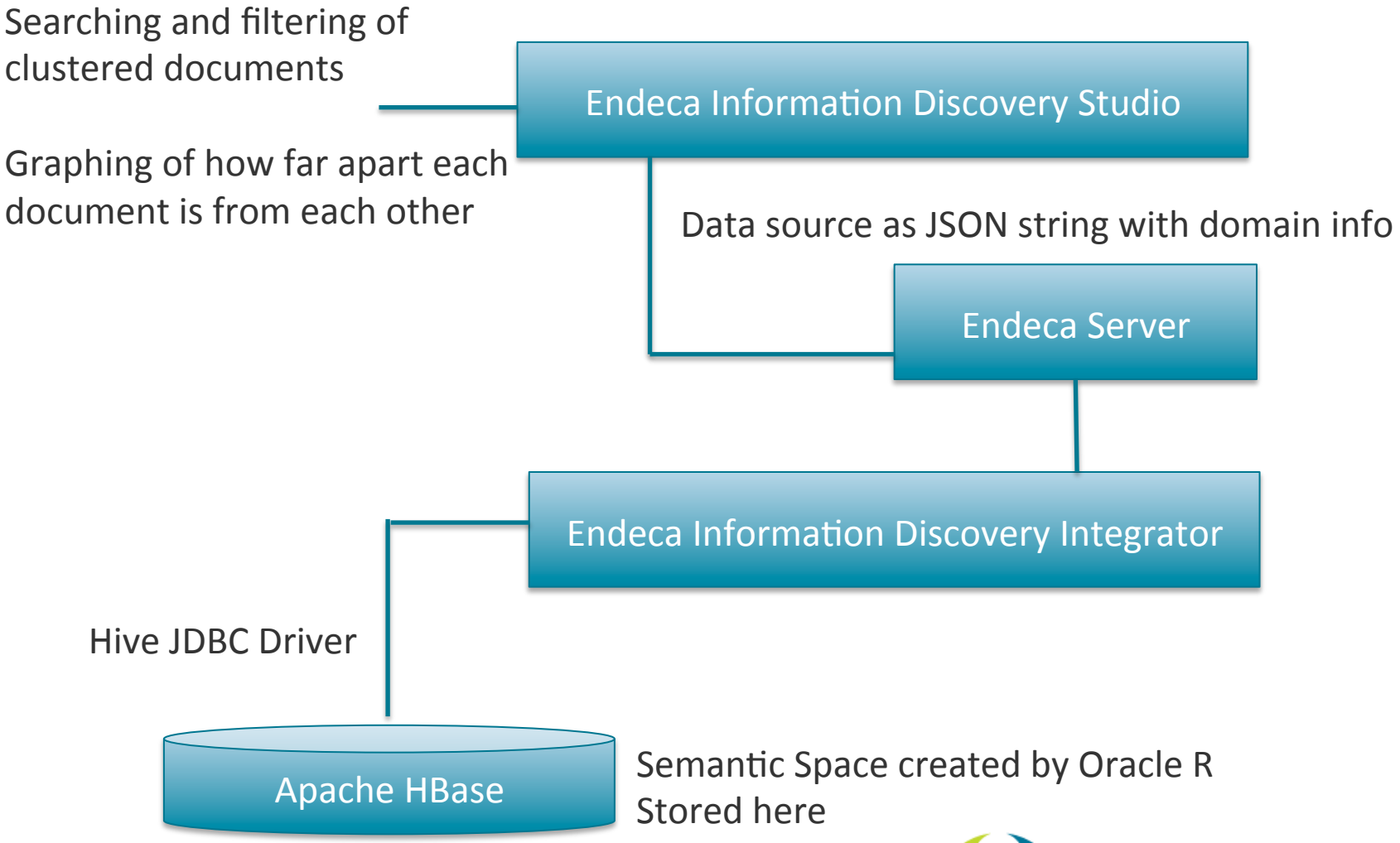
	Document 1	Document 2	Document 3	.....
Document 2	6.9873			
Document 3	8.1201	17.9188		
Document 4	9.6577	2.14327	2.2945	

.....

Persisting the Semantic Space

- ORE Functions for Hive are used again to save results from R text mining in Apache HBase.
- HBase schema includes one Row for each Document.
- For each row, there are two Column Families.
- First Column Family has one pair of Key=<Document link>, Value=<Original, unprocessed document>
- Second Column Family has multiple pairs of Key=<semantic term>, Value=<word frequency> derived for that Document via text mining in R.

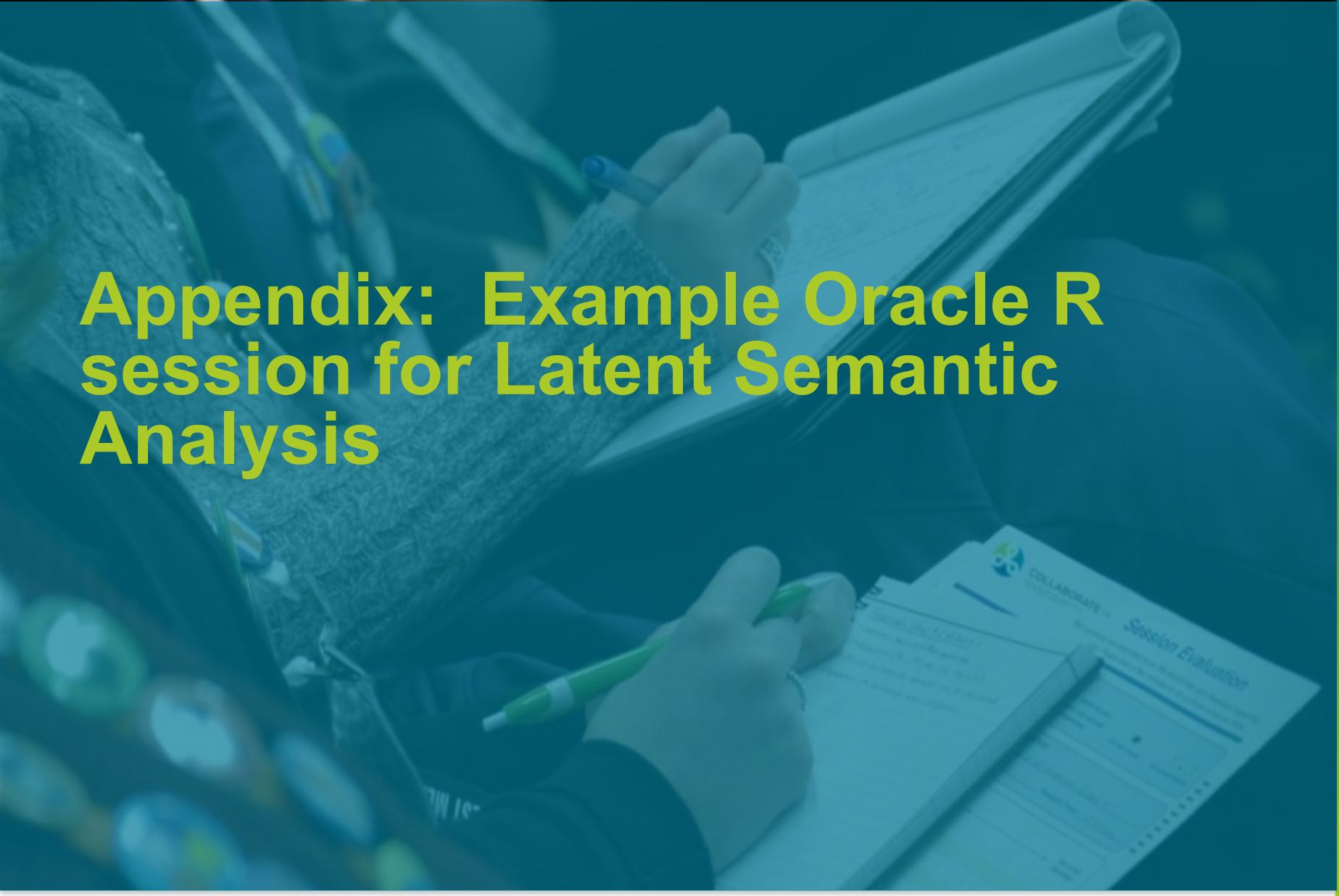
Connecting Endeca to the Semantic Space



Lessons Learned

- Pull-based scheduling for the Hadoop Task Tracker leads to cluster underutilization during peak loads.
- A centralized management interface similar to Cloudera's offerings in the Oracle Big Data Appliance is essential.
- Aggregating multiple HDFS files into a matrix in R is slow when using either RHIPE or ORCH.
- The foundation for scaling to the use of an Oracle Big Data Appliance connected to Oracle Exalytics has been set.

Appendix: Example Oracle R session for Latent Semantic Analysis



LSA and Singular Value Decomposition (SVD)

- A mathematical technique useful for easing the difficulty of comparing multiple documents to each other.
- Rows represent documents, Columns represent words.
- SVD reduces the number of columns while preserving the similarity structure across the rows.
- Documents are then compared by taking the cosine between the vectors for any two rows.
- Documents similar to each other will have their cosine close to 1 and those dissimilar will their cosine close to 0.

Preparing Oracle R for LSA

```
[root@samantha Downloads]# R CMD INSTALL slam_0.1-32.tar.gz
[root@samantha Downloads]# R CMD INSTALL tm_0.5-10.tar.gz
[root@samantha Downloads]# yum install libxml2
[root@samantha Downloads]# yum install libxml2-devel
[root@samantha Downloads]# R CMD INSTALL XML_3.98-1.1.tar.gz
[root@samantha Downloads]# R CMD INSTALL SnowballC_0.5.tar.gz
[root@samantha Downloads]# R CMD INSTALL lsa_0.73.tar.gz
[root@samantha Downloads]# R CMD INSTALL scatterplot3d_0.3-35.tar.gz
[root@samantha Downloads]# R CMD INSTALL ggplot2_0.9.3.1.tar.gz
[root@samantha Downloads]# R

> setRepositories()
> install.packages("ggplot2", dependencies=TRUE)
> library(tm)
> library(slam)
> library(lsa)
Loading required package: SnowballC
> library(ggplot2)
> library(scatterplot3d)
```

Using NLP to Create a Corpus in Oracle R

```
>  reuters <- Corpus(DirSource(reut21578),
                     readerControl = list(reader = readReut21578XML))
> reuters <- tm_map(reuters, as.PlainTextDocument)
> reuters <- tm_map(reuters, tolower)
> reuters <- tm_map(reuters, removeWords, stopwords("english"))
> reuters <- tm_map(reuters, removePunctuation)
> reuters <- tm_map(reuters, stemDocument)
> reuters <- tm_map(reuters, removeNumbers)
> reuters <- tm_map(reuters, stripWhitespace)
> reuters
A corpus with 20 text documents
```

Creating a Corpus in Oracle R

```
> termMat <- as.matrix(TermDocumentMatrix(reuters))
> termMat.lsa <- lw_logtf(termMat) * gw_idf(termMat)
```

$$idf(i,j) = \begin{cases} 0 & \text{if } wf_{i,j} = 0 \\ (1 + \log(wf_{i,j})) \log \frac{N}{df_i} & \text{if } wf_{i,j} \geq 1 \end{cases}$$

```
> semanticSpace <- lsa(termMat.lsa)
> semanticSpace.dist <- dist(t(as.textmatrix(semanticSpace)))
```

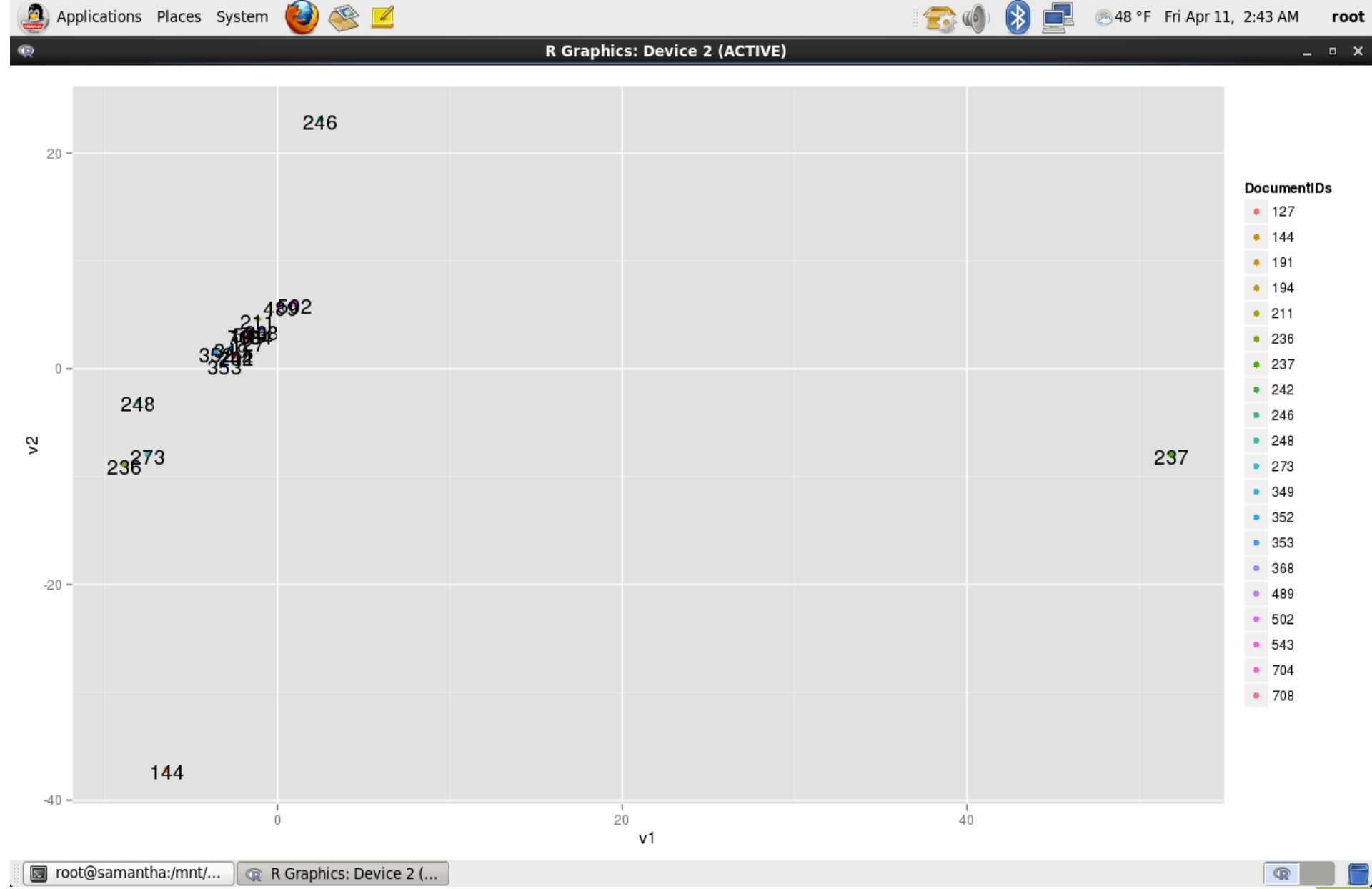
Comparing the Documents in Space

Scaled down to 2-D Space:

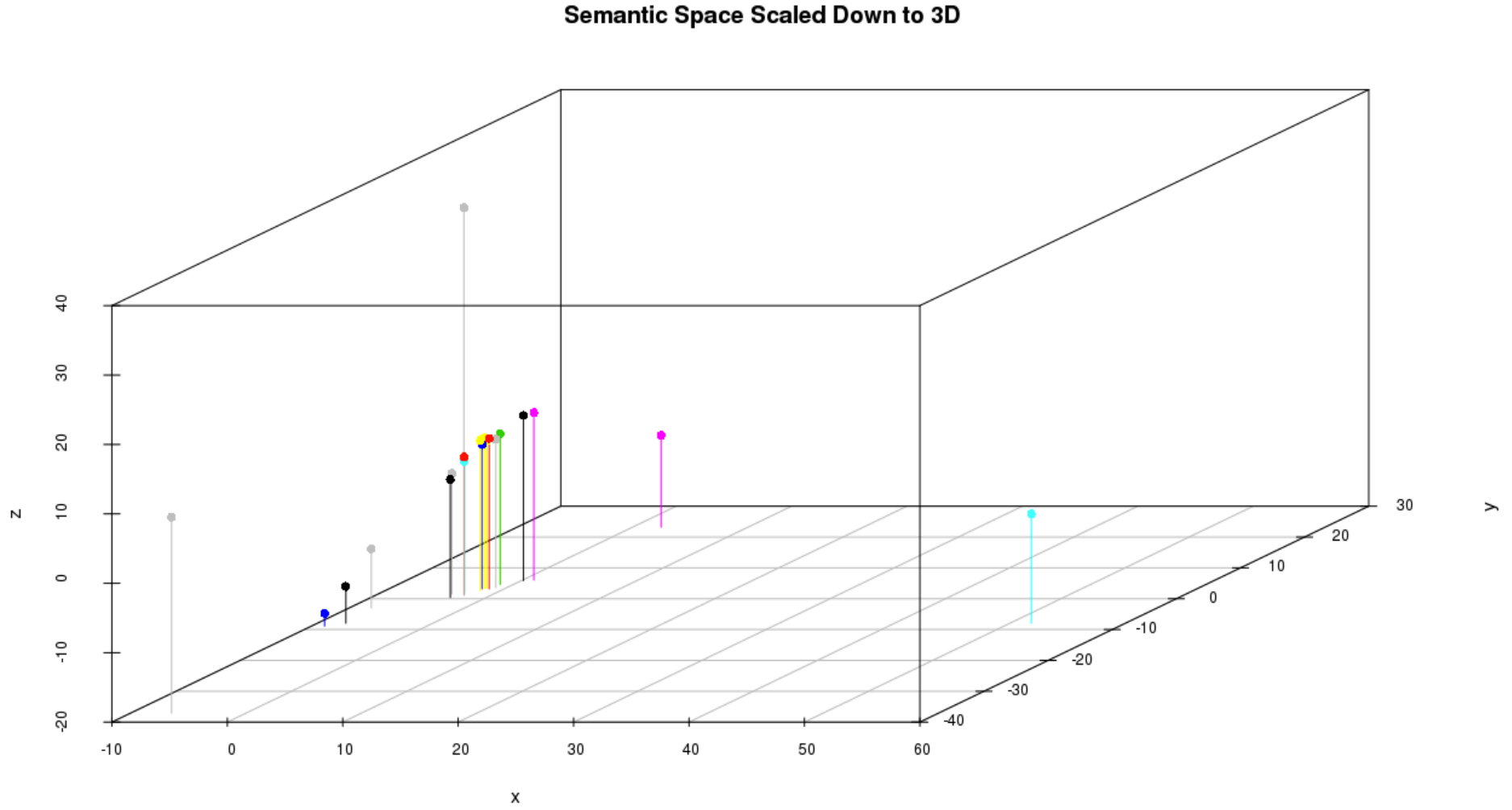
```
> fit <- cmdscale(semanticSpace.dist, eig=TRUE, k=2)
> fit.scaled <- data.frame(v1=fit$points[, 1], v2=fit$points[, 2])
> DocumentIDs <- row.names(fit$points)
> qplot(data=fit.scaled,x=v1,y=v2,color=DocumentIDs)
> ggplot(fit.scaled, aes(x=v1, y=v2)) + geom_point(data=fit.scaled,
        aes(x=v1, y=v2, color=DocumentIDs)) +
        geom_text(data=fit.scaled, aes(x=v1, y=v2-0.2, label=DocumentIDs)
```

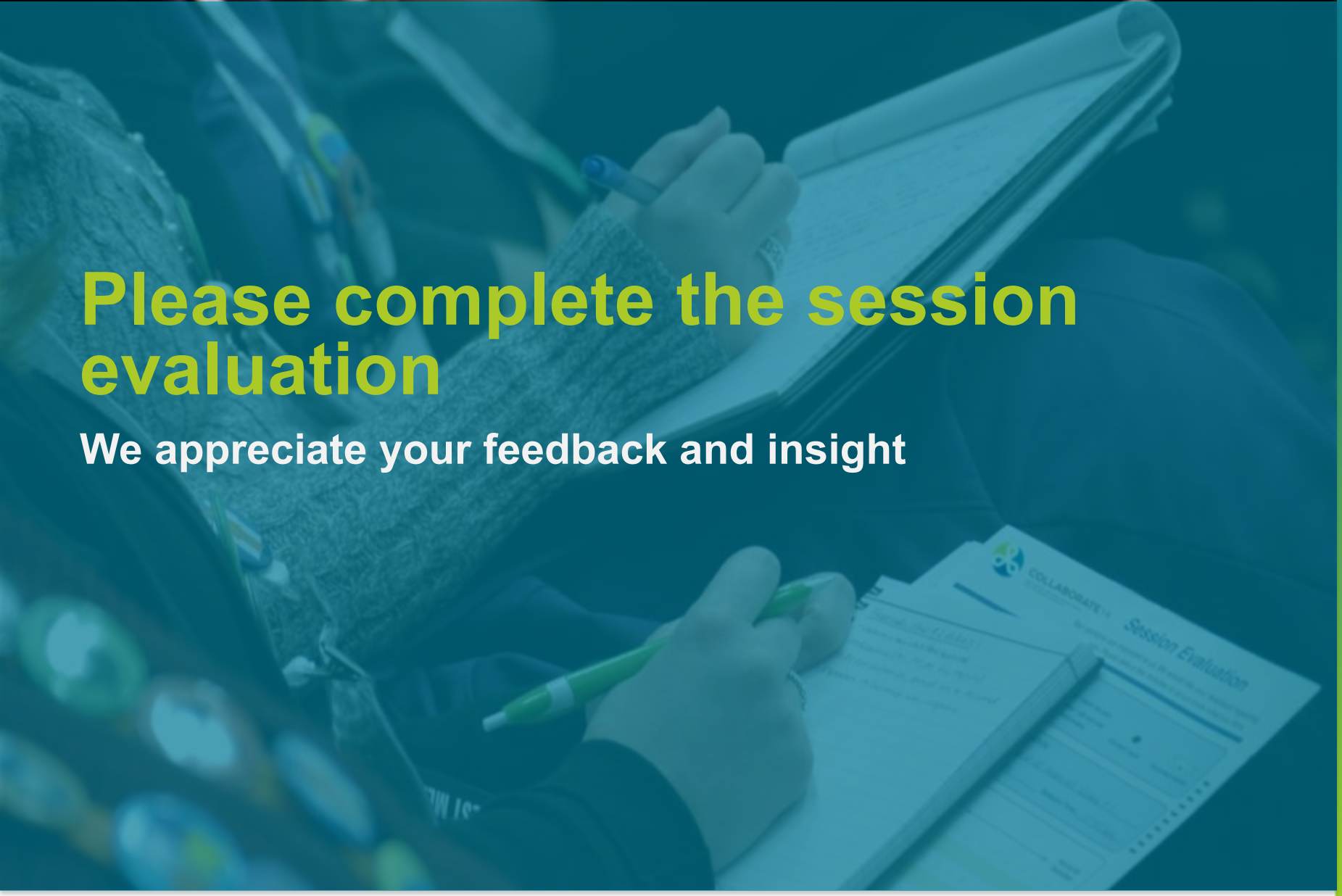
Scaled down to 3-D Space:

```
> fit <- cmdscale(semanticSpace.dist, eig=TRUE, k=3)
> scatterplot3d(fit$points[,1], fit$points[,2], fit$points[,3], color=DocumentIDs,
                pch=16, main="Semantic Space Scaled Down to 3D", xlab="x",
                ylab="y", zlab="z", type="h")
```


Applications Places System
48 °F Fri Apr 11, 2:43 AM root
R Graphics: Device 2 (ACTIVE)
246
248
273
236
237
144
353
DocumentIDs
127
144
191
194
211
236
237
242
246
248
273
349
352
353
368
489
502
543
704
708
v1
v2
20
0
-20
-40
0
20
40
root@samantha:/mnt/...
R Graphics: Device 2 (...

R Graphics: Device 2 (ACTIVE)





Please complete the session evaluation

We appreciate your feedback and insight

You may complete the session evaluation either on paper or online via the mobile app

COLLABORATE14
TECHNOLOGY AND APPLICATIONS FORUM FOR THE ORACLE COMMUNITY

#C14LV