SAS ARGENTINA

16 DE OCTUBRE DE 2014

¿QUE ES BIG DATA ANALYTICS? IMPORTANCIA E IMPACTO SAS SOBRE HADOOP

Sergio Uassouf

Líder de Práctica de
Gestión de Información e Infraestructura

v4.3 20141015

SAS NUESTRA EMPRESA

"With no shareholders demanding short-term returns, we are free to invest in a sustainable future. That's why we invest in a dedicated workforce, sustainable operations and a strong community – to make everyone, not just SAS, successful." –Jim Goodnight

SOLUCIONES ANALITICAS "LLAVE EN MANO"

HERRAMIENTAS PARA DESARROLLOS ANALITICOS

BIG DATA ¿MODA O REALIDAD PERDURABLE?

FALSO DILEMA ¿NEGOCIOS O TI?

PERO SE VE MUY FRECUENTEMENTE... CASI SIEMPRE

- ¿Un negocio que no tenga soporte tecnológico?

- ¿O una tecnología que no sirva para generar negocios?

- También conocido como Sindrome de la Gata Flora

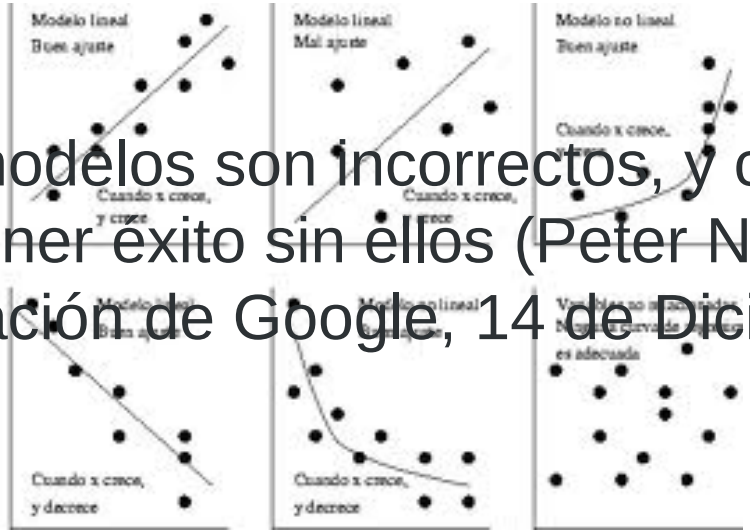
HITOS IMPORTANTES DE LA INFORMATICA

- La multiprogramación y el spool de impresoras.
- Los monitores de transacciones y el procesamiento online.
- Las bases de datos relacionales.
- La programación orientada a objetos.
- Una computadora en cada escritorio.
- El protocolo IP = Internet.
- El protocolo XML = HTML = World Wide Web.
- Google.
- *¿Big Data?.*

He buscado adjetivos que no resulten ofensivos, pero no los encontré.

¿INTELIGENCIA O FUERZA BRUTA?

- Todos los modelos son incorrectos, pero algunos son útiles (George Box, estadístico, 18 de Octubre de 1919 – 28 de Marzo de 2013).

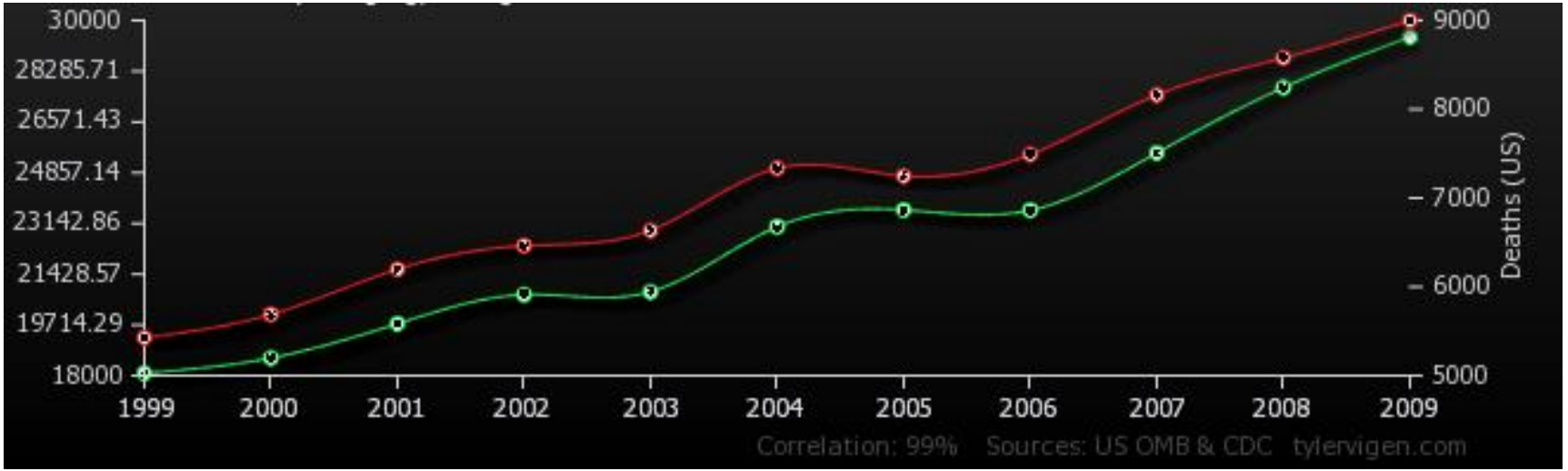


- Todos los modelos son incorrectos, y cada vez más podemos tener éxito sin ellos (Peter Norvig, director de investigación de Google, 14 de Diciembre de 1956).

CUIDADO CON LAS APARIENCIAS

CORRELACION NO SIGNIFICA NECESARIAMENTE CAUSALIDAD

- Gasto de U.S.A. en ciencia y tecnología...
  Correlación 0.992082 con...
- Suicidios por estrangulamiento, ahorcamiento y sofocación

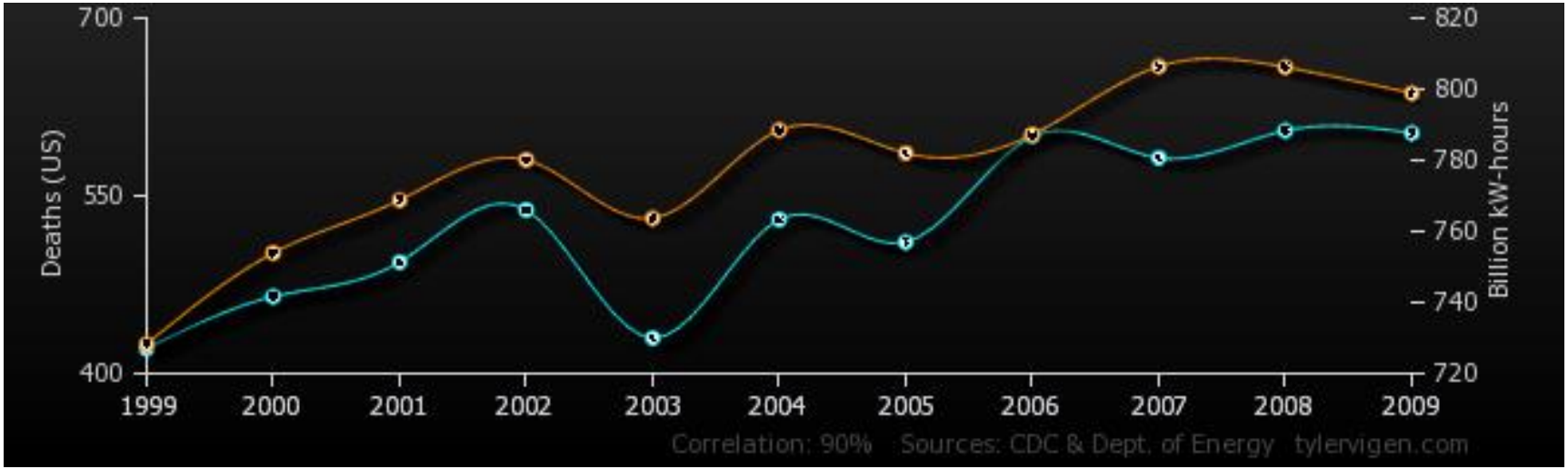


Fuente: Spurious Correlations; http://www.tylervigen.com/

CUIDADO CON LAS APARIENCIAS

CORRELACION NO SIGNIFICA NECESARIAMENTE CAUSALIDAD

- Cantidad de gente que se ahoga nadando en una pileta...
  Correlación 0.901179 con...
- Potencia generada por las plantas nucleares

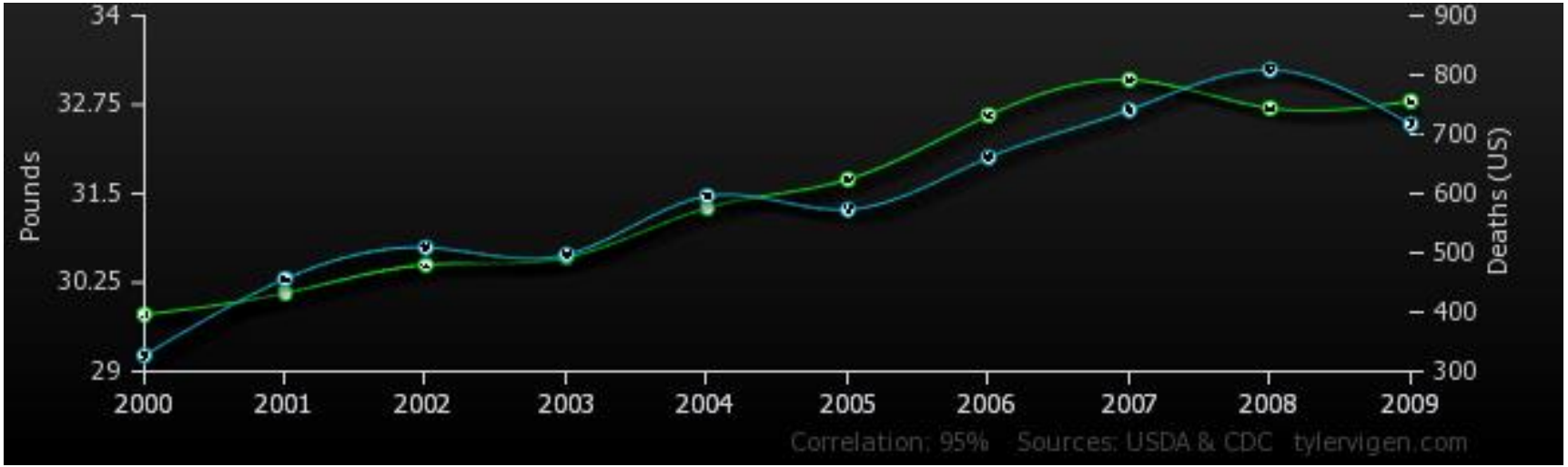


Fuente: Spurious Correlations; http://www.tylervigen.com/

CUIDADO CON LAS APARIENCIAS

CORRELACION NO SIGNIFICA NECESARIAMENTE CAUSALIDAD

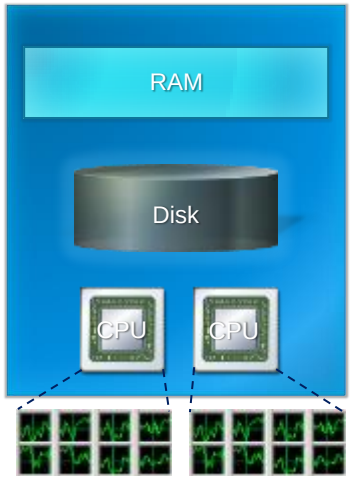
- Consumo de queso per capita...
  Correlación 0.947091 con...
- Muertes por enredo en la ropa de cama



Fuente: Spurious Correlations; http://www.tylervigen.com/

COMPONENTES BÁSICOS HACE 40 AÑOS

- Desde los inicios de la informática un computador, ya sea personal o empresarial está compuesto de 3 componentes principales.



MEMORIA

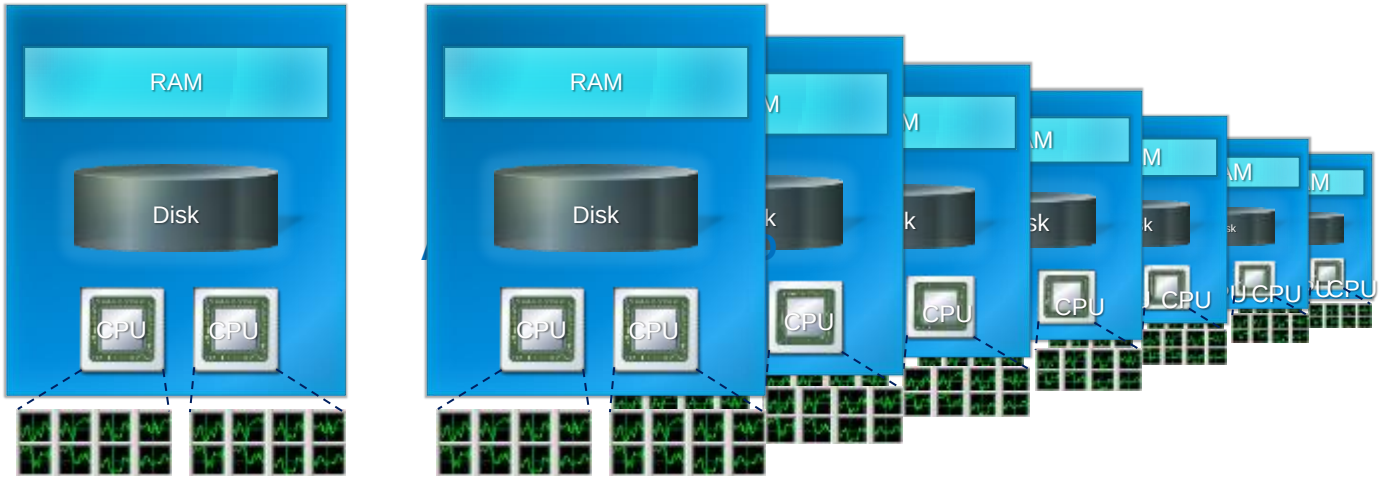
UNIDADES DE ALMACENAMIENTO

UNIDADES DE PROCESAMIENTO

PONIENDONOS EN CONTEXTO

COMPONENTES BÁSICOS EN LA ACTUALIDAD

- Desde los inicios de la informática hasta la actualidad, todo sistema computacional personal o empresarial está compuesto de 3 componentes principales.



- Proyecto Durkheim
- Craig Venter, Celera Genomics
- eBay
  - *Aproximadamente 15 terabytes.*
  - *200.000.000 grabaciones por día.*
  - *Importaciones masivas (bulk load): 500.000.000 de filas en 30 minutos.*
  - *1,2 terabytes importados cada día.*
- Facebook
  - *Almacenamiento de mensajes y sus índices.*
  - *75.000.000.000 de lecturas/grabaciones por día.*
  - *En momentos pico 1.500.000 de operaciones por segundo.*
  - *2 petabytes en HBase.*

- Proyecto Durkheim (... o La vida es bella)
- Craig Venter, Celera Genomics (... ACGT, Anibal Troilo / Carlos Gardel)
- eBay
  - *Aproximadamente 15 terabytes.*
  - *200.000.000 grabaciones por día.*
  - *Importaciones masivas (bulk load): 500.000.000 de filas en 30 minutos.*
  - *1,2 terabytes importados cada día.*
- Facebook
  - *Almacenamiento de mensajes y sus índices.*
  - *75.000.000.000 de lecturas/grabaciones por día.*
  - *En momentos pico 1.500.000 de operaciones por segundo.*
  - *2 petabytes en HBase.*

HADOOP | TAMAÑOS EN PERSPECTIVA

TRANSACCIONES BANCARIAS	
Bytes / Transacción	Cien
Bytes en 1 TB	Un billón
Transacciones en 1 TB	Diez mil millones
A 300 Txs / Segundo	4 años de transacciones de días hábiles de 8 horas

- Cm ands?
- Pr / ojt

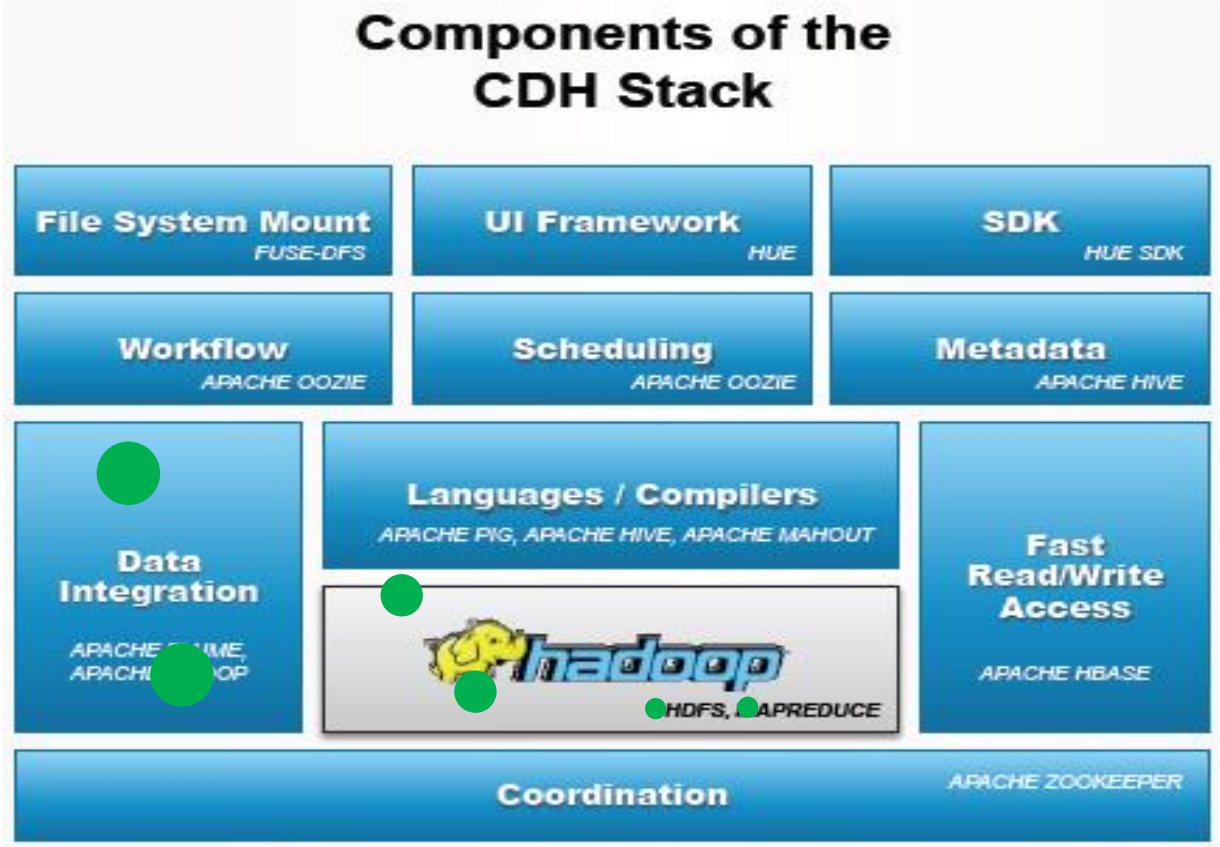
SHORT MESSAGES SERVICES	
Tamaño máximo	Ciento sesenta caracteres
Tamaño promedio	25
SMS promedio en 1 TB	Cuarenta mil millones

¿20 minutos?

"ECOSISTEMA" HADOOP (COMPONENTES)

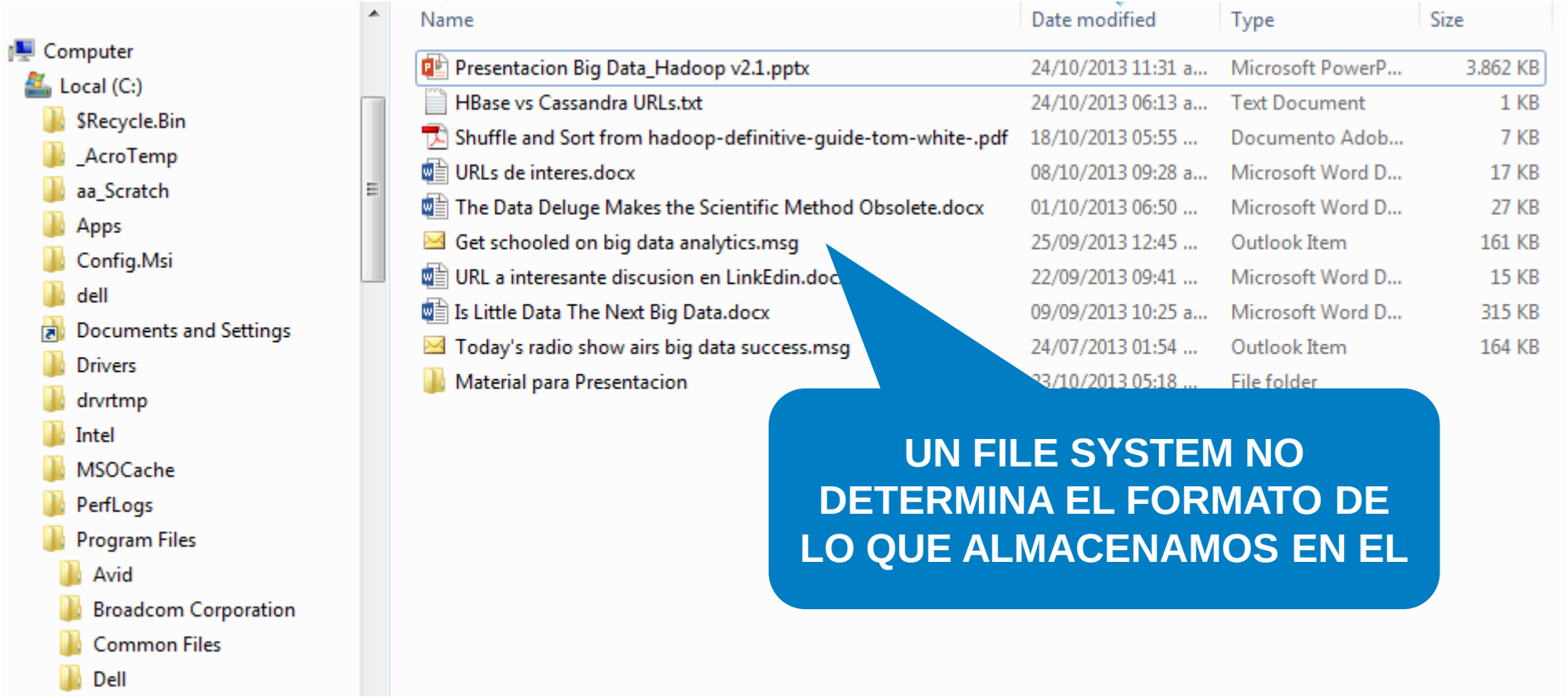
SISTEMA DE ARCHIVOS HDFS

MODELO DE PROGRAMACION MAP/REDUCE (Y OTROS)



- Entonces Big Data...
- Significa **Procesamiento Masivamente Paralelo** (MPP)...
  - ¿Big Data necesariamente es Hadoop?.
  - ¿Big Data es necesariamente HDFS?
  - ¿Big Data es necesariamente MapReduce?

DIGRESION ¿QUE ES UN FILE SYSTEM?



UN FILE SYSTEM NO DETERMINA EL FORMATO DE LO QUE ALMACENAMOS EN EL

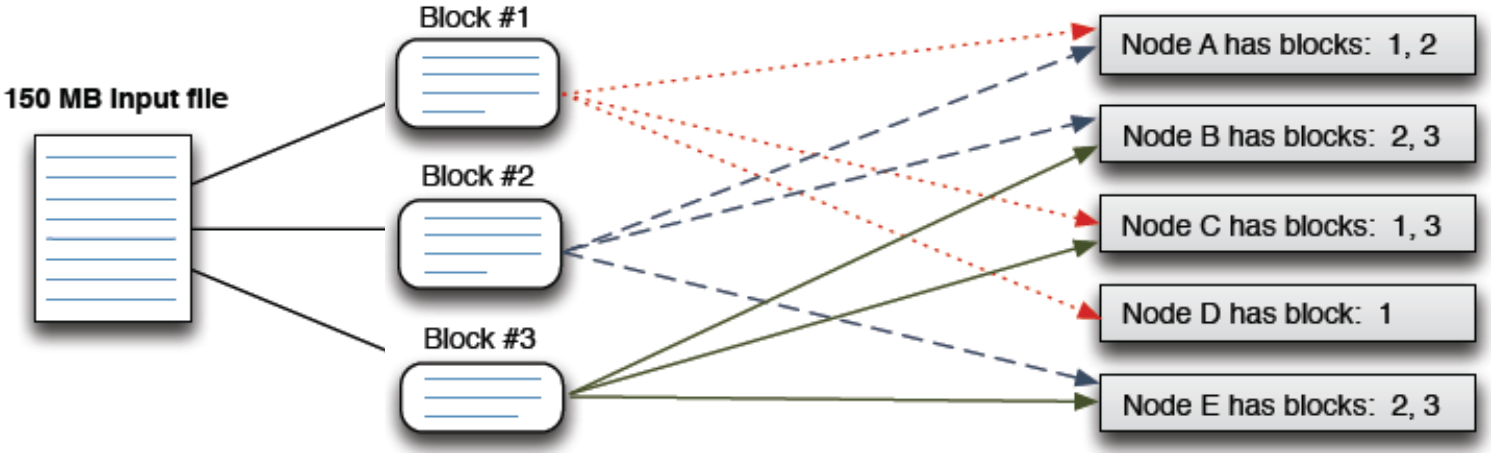
PREMISAS FUNDAMENTALES DE DISEÑO

- Concepto central: Distribuir los datos inicialmente a medida que van siendo almacenados en el sistema.
  - *Cada nodo trabaja con los datos locales de ese nodo.*
  - *Los datos son replicados múltiples veces en diferentes nodos.*
  - *Los cálculos ocurren donde sea que están almacenados los datos.*

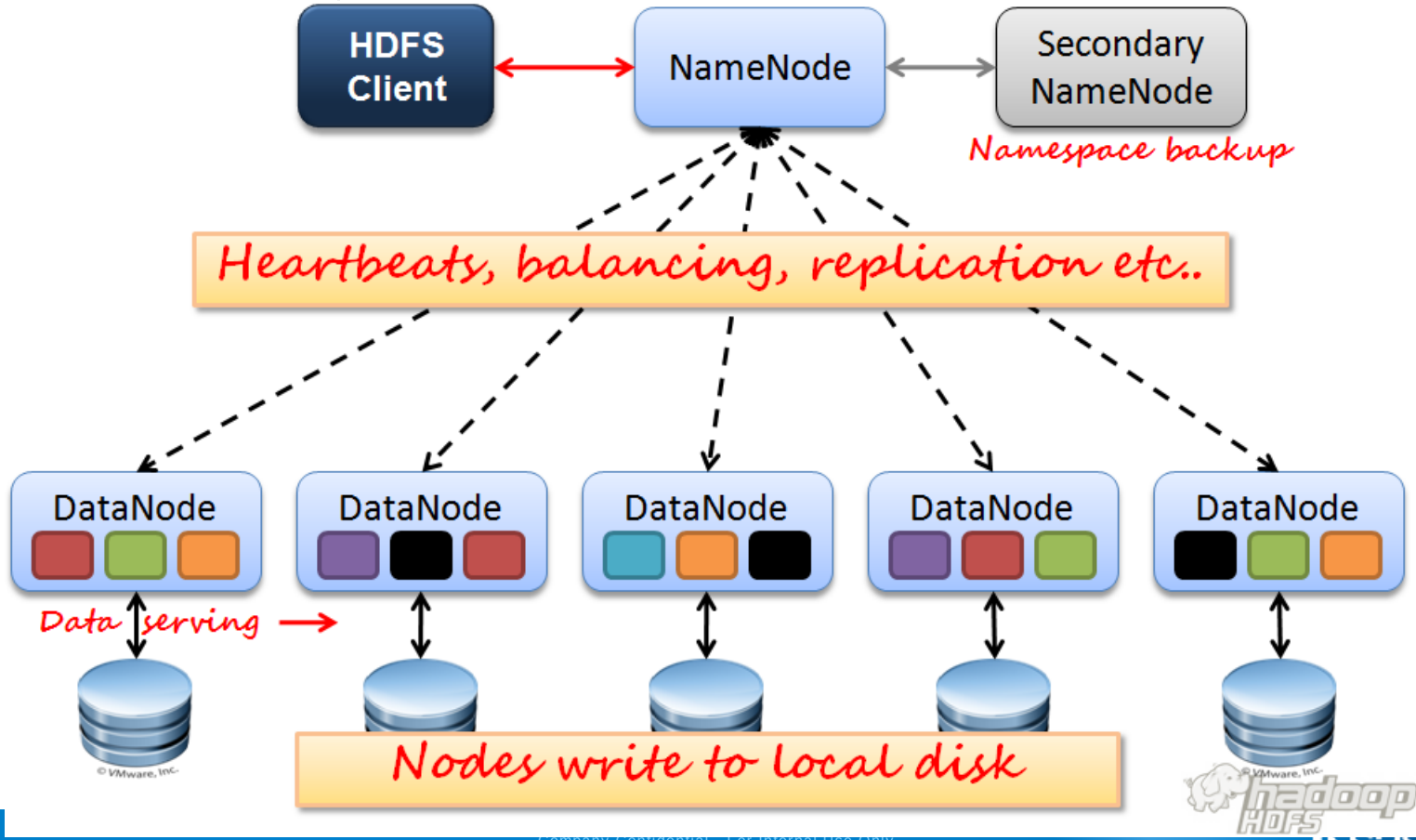
- Los nodos intercambian entre ellos la menor cantidad de información posible.
  - *Arquitectura “shared nothing”.*

UN PAR DE ESQUEMAS GRAFICO Y SEGUIMOS...

- Los bloques son replicados en los nodos componentes del cluster.
  - *Basados en un factor de replicación (por defecto 3).*
- La replicación no sólo mejora la disponibilidad, sino la performance.
  - *Mayores oportunidades para conseguir datos locales.*



HDFS OTRO ESQUEMA GRAFICO



MAS PREMISAS DE DISEÑO HADOOP

- Almacenamiento redundante para volúmenes masivos de información, previendo alta cantidad de fallas de los componentes.
  - *Utilizando hardware commodity que tienden a fallar frecuentemente.*
- Basado en Google File System.
  - *Diseñado para archivos terabytes o petabytes.*
- Enormes flujos de lecturas secuenciales.
  - *Favorece un throughput muy elevado y sostenido sobre la baja latencia.*
  - *No hay acceso random competitivo contra los métodos tradicionales (transaccionales). Muchos proyectos en investigación.*

HADOOP/MAP-REDUCE Y ECOSISTEMA

ALMACENAR Y ANALIZAR GRANDES VOLUMENES DE INFORMACION A BAJO COSTO

- Si puede almacenar mucha más información a un costo mucho menor...
- Y puede procesarla en un tiempo mucho menor.
- Entonces no necesita armar modelos tomando sólo un subconjunto de los datos...
- Y puede hacer todas las iteraciones que necesite.
- **Entonces puede almacenar y procesar la información que antes no podía**

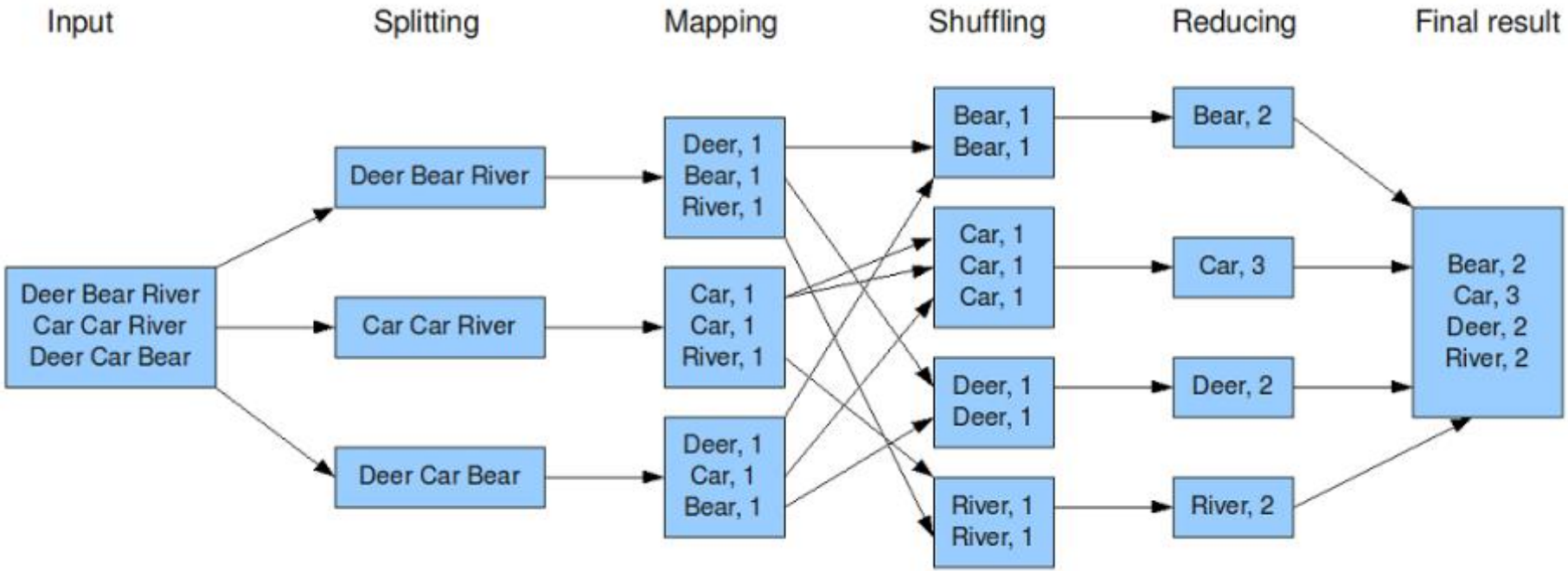
NECESIDAD A RESOLVER

ALMACENAR Y ANALIZAR GRANDES VOLUMENES DE INFORMACION A BAJO COSTO

MAP_REDUCE

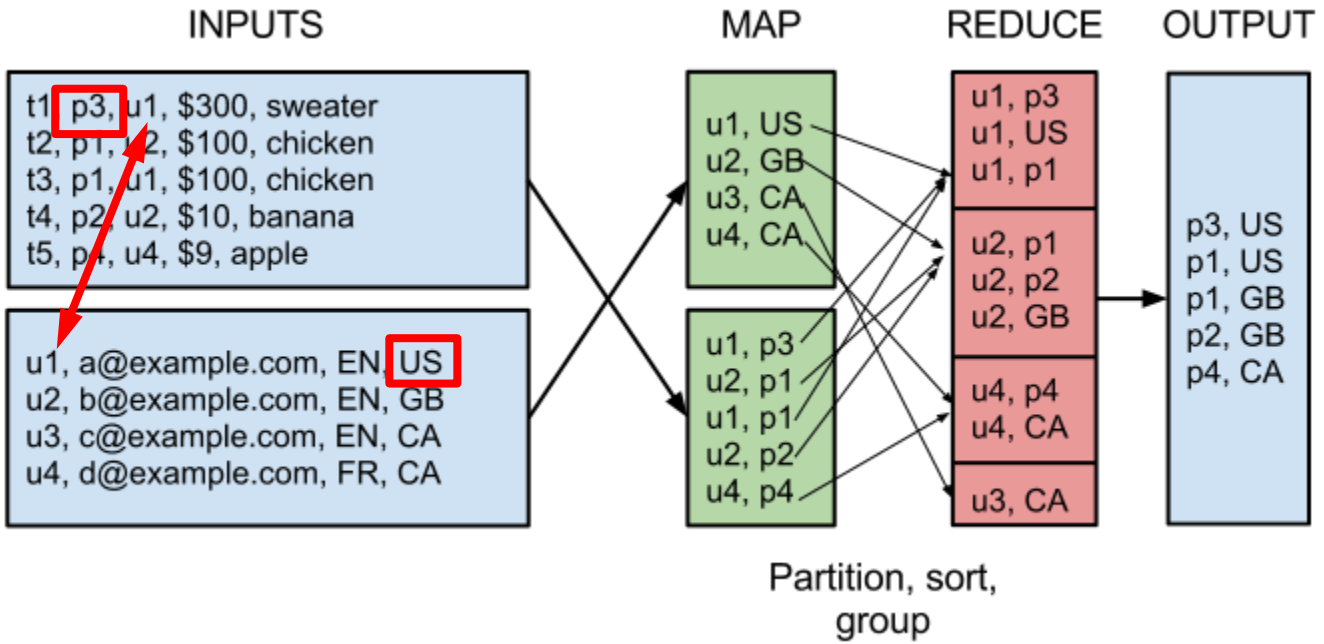
Hay 10 tipos
de personas,
las que entienden
los números binarios
y las que no

MAP_REDUCE PROCESO DE CONTEO DE PALABRAS



MAP_REDUCE PROCESO DE JOIN SQL

- *Apareando la variable 3 del Input A con la variable 1 del Input B...*
- *Informar la variable 2 del Input A y la variable 4 del input B.*



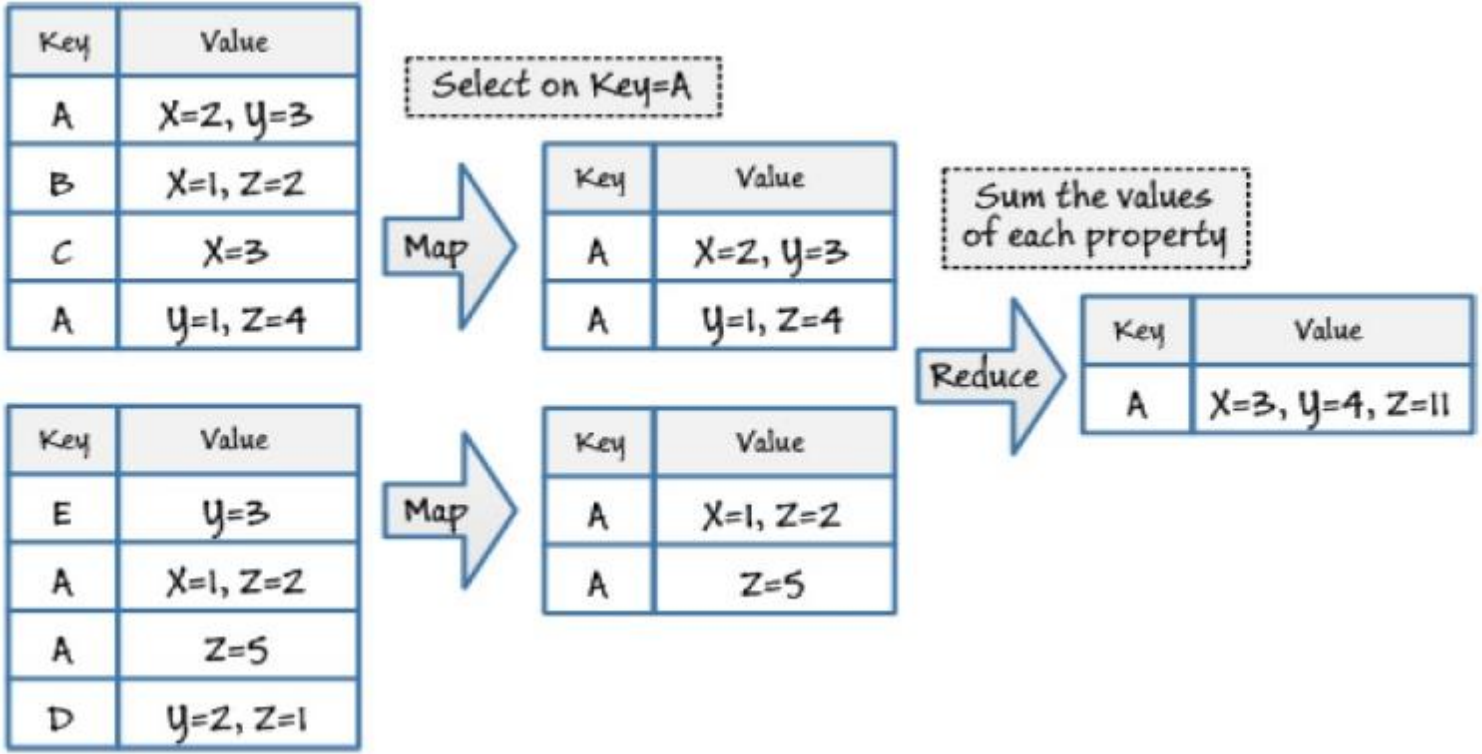
MAP_REDUCE

SOLO PARA ASUSTAR... (DESPUES PASARÁ)
PROGRAMA MAP REDUCE DE SENTENCIA JOIN SQL

```
public class JoinStationMapper extends MapReduceBase implements
Mapper<LongWritable, Text, TextPair, Text> {
      private NcdcStationMetadataParser parser = new
NcdcStationMetadataParser();

      public void map(LongWritable key, Text value,
OutputCollector<TextPair, Text> output, Reporter reporter)
throws IOException {
            if (parser.parse(value)) {
              output.collect(new TextPair(parser.getStationId(),
"0"), new Text(parser.getStationName()));
            }
      }
}

public class JoinRecordMapper extends MapReduceBase implements
Mapper<LongWritable, Text, TextPair, Text> {
      private NcdcRecordParser parser = new NcdcRecordParser();
      public void map(LongWritable key, Text value,
OutputCollector<TextPair, Text> output, Reporter reporter)
throws IOException {
      parser.parse(value); output.collect(new
TextPair(parser.getStationId(), "1"), value);
      }
}

public class JoinReducer extends MapReduceBase implements
Reducer<TextPair, Text, Text, Text> {
      public void reduce(TextPair key, Iterator<Text> values,
OutputCollector<Text, Text> output, Reporter reporter) throws
IOException {
            Text stationName = new Text(values.next());
            while (values.hasNext()) {
                  Text record = values.next();
                  Text outValue = new Text(stationName.toString()
+ "\t" + record.toString());
                  output.collect(key.getFirst(), outValue);
            }
      }
}

public class JoinRecordWithStationName extends Configured
implements Tool {
      public static class KeyPartitioner implements
Partitioner<TextPair, Text> {
            @Override
            public void configure(JobConf job) {}

      @Override
      public int getPartition(TextPair key, Text value, int
numPartitions) {
            return (key.getFirst().hashCode() &
Integer.MAX_VALUE) % numPartitions;
            }
      }
      @Override
      public int run(String[] args) throws Exception {
            if (args.length != 3) {
                  JobBuilder.printUsage(this, "<ncdc input>
<station input> <output>");
                  return -1;
            }
            JobConf conf = new JobConf(getConf(), getClass());
            conf.setJobName("Join record with station name");
            Path ncdcInputPath = new Path(args[0]);
            Path stationInputPath = new Path(args[1]);
            Path outputPath = new Path(args[2]);
            MultipleInputs.addInputPath(conf, ncdcInputPath,
TextInputFormat.class, JoinRecordMapper.class);
            MultipleInputs.addInputPath(conf, stationInputPath,
TextInputFormat.class, JoinStationMapper.class);
            FileOutputFormat.setOutputPath(conf, outputPath);
            conf.setPartitionerClass(KeyPartitioner.class);

      conf.setOutputValueGroupingComparator(TextPair.FirstCompa
rator.class);
            conf.setMapOutputKeyClass(TextPair.class);
            conf.setReducerClass(JoinReducer.class);
            conf.setOutputKeyClass(Text.class);
            JobClient.runJob(conf);
            return 0;
      }

      public static void main(String[] args) throws Exception {
            int exitCode = ToolRunner.run(new
JoinRecordWithStationName(), args);
            System.exit(exitCode);
      }
}
```

MAP_REDUCE PROCESO DE SUMA DE VARIABLES

- *Informar la suma de las variables del Input A y B cuyas claves apareen.*



MAP_REDUCE ENCADENAMIENTO DE PROGRAMAS MAP_REDUCE

- *Hay que diseñar el programa pensando en el paralelismo (analogía OOP).*

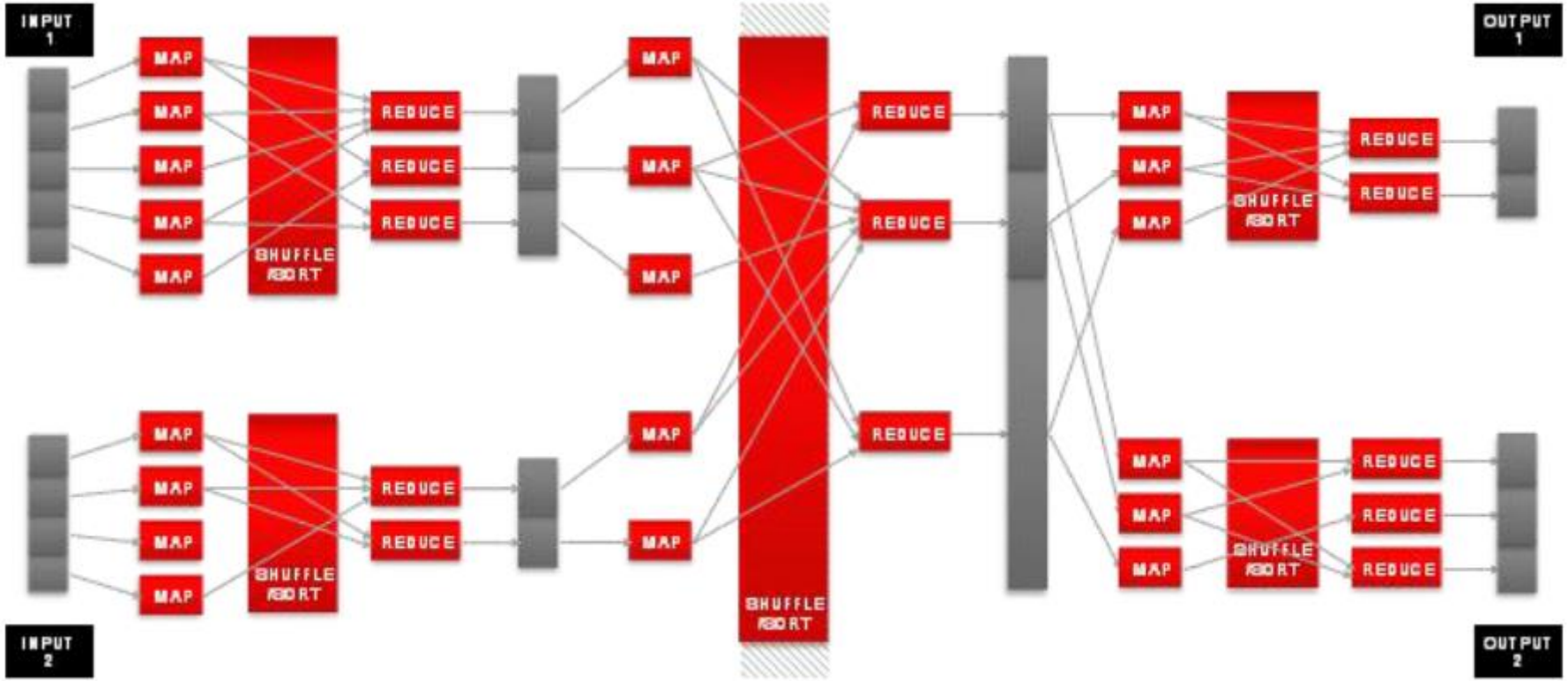
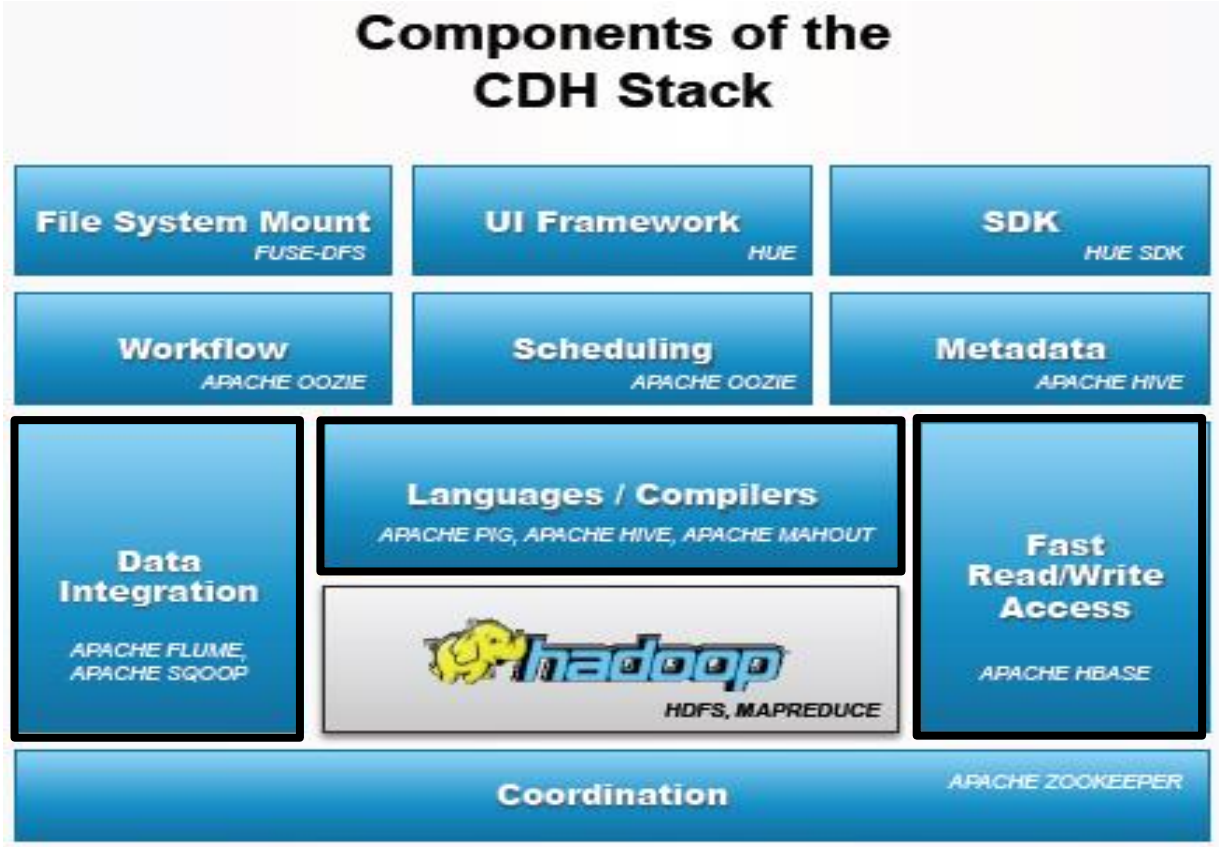


Gráfico tomado de Oracle

“ECOSISTEMA” HADOOP (COMPONENTES)

- Apache Hive
- Apache Pig
- Apache HBase
- Apache Sqoop
- Apache Flume
- Apache Mahout



JAVA MAP_REDUCE VS HIVE

```
public class JoinStationMapper extends MapReduceBase implements
Mapper<LongWritable, Text, TextPair, Text> {
     private NcdcStationMetadataParser parser = new
NcdcStationMetadataParser();

     public void map(LongWritable key, Text value,
OutputCollector<TextPair, Text> output, Reporter reporter)
throws IOException {
          if (parser.parse(value)) {
            output.collect(new TextPair(parser.getStationId(),
"0"), new Text(parser.getStationName()));
          }
     }
}
```

```
     @Override
     public int getPartition(TextPair key, Text value, int
numPartitions) {
          return (key.getFirst().hashCode() &
Integer.MAX_VALUE) % numPartitions;
          }
     }
     @Override
     public int run(String[] args) throws Exception {
          if (args.length != 3) {
               JobBuilder.printUsage(this, "<ncdc input>
<station input> <output>");
               return -1;
          }
```

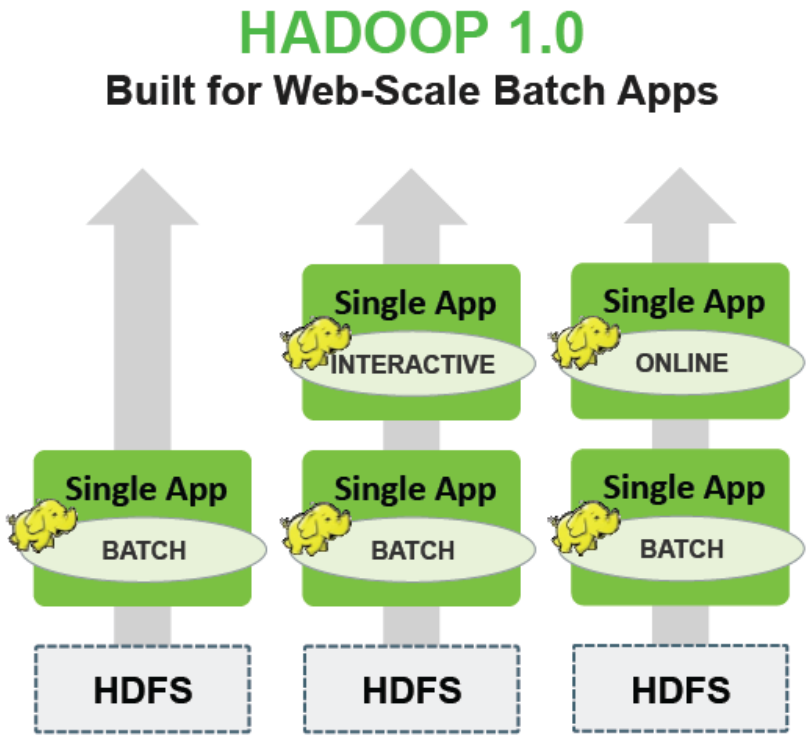
SELECT * FROM Stations JOIN Records ON (Stations.StationID = Records.StationID);

```
TextPair(parser.getStationId(), "1"), value);
     }
}

public class JoinReducer extends MapReduceBase implements
Reducer<TextPair, Text, Text, Text> {
     public void reduce(TextPair key, Iterator<Text> values,
OutputCollector<Text, Text> output, Reporter reporter) throws
IOException {
          Text stationName = new Text(values.next());
          while (values.hasNext()) {
               Text record = values.next();
               Text outValue = new Text(stationName.toString()
+ "\t" + record.toString());
               output.collect(key.getFirst(), outValue);
          }
     }
}

public class JoinRecordWithStationName extends Configured
implements Tool {
     public static class KeyPartitioner implements
Partitioner<TextPair, Text> {
          @Override
          public void configure(JobConf job) {}
```

```
TextInputFormat.class, JoinStationMapper.class);
          FileOutputFormat.setOutputPath(conf, outputPath);
          conf.setPartitionerClass(KeyPartitioner.class);

     conf.setOutputValueGroupingComparator(TextPair.FirstCompa
rator.class);
          conf.setMapOutputKeyClass(TextPair.class);
          conf.setReducerClass(JoinReducer.class);
          conf.setOutputKeyClass(Text.class);
          JobClient.runJob(conf);
          return 0;
     }

     public static void main(String[] args) throws Exception {
          int exitCode = ToolRunner.run(new
JoinRecordWithStationName(), args);
          System.exit(exitCode);
     }
}
```

PARTE 3
HADOOP VERSION 2

HADOOP 1.0 – REQUERIMIENTOS PARA HADOOP 2.0



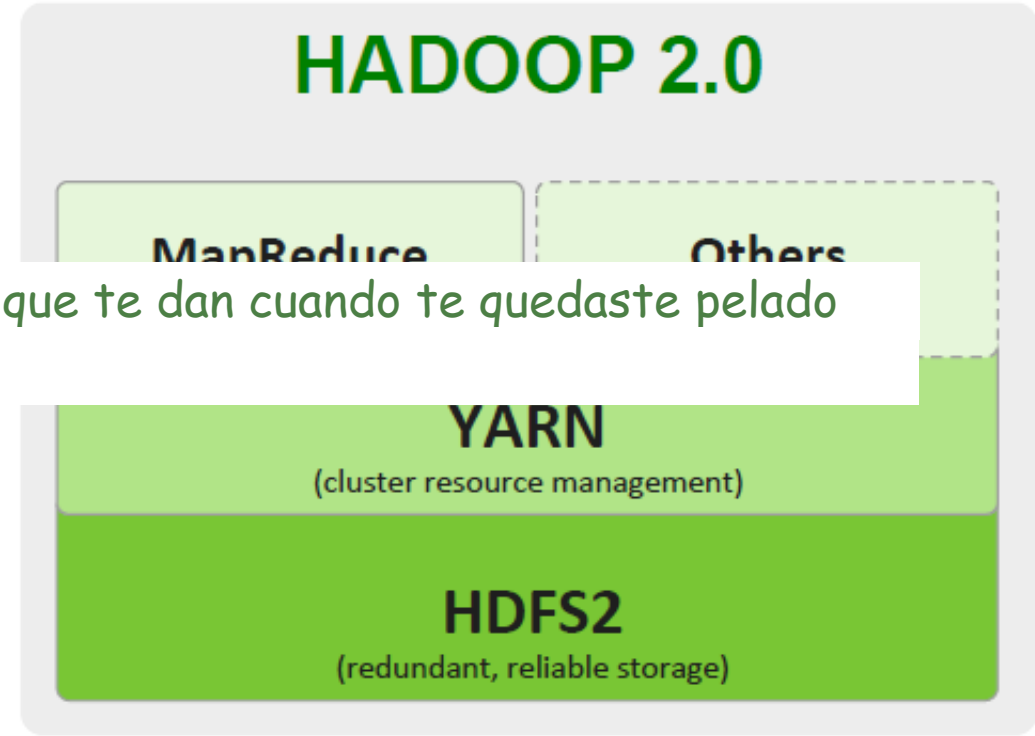
- Alta disponibilidad para el NameNode HDFS.
- NameNode federado para mayor escalabilidad.
- Acceso NFS para montar HDFS como un file system estándar.
- Encripción de datos en tránsito.
- Sistema YARN de administración de recursos.
- Separación de HDFS respecto al modelo de programación MapReduce.

Gráfico tomado de Hortonworks

HADOOP | HADOOP 2.0

Único Uso
Aplicaciones Batch

Plataforma multi-propósito
Batch, Interactivo, Online, Streaming



La experiencia es un peine que te dan cuando te quedaste pelado
Oscar "Ringo" Bonavena

Gráfico tomado de Hortonworks

HADOOP 2.0 – MODELOS DE PROGRAMACION

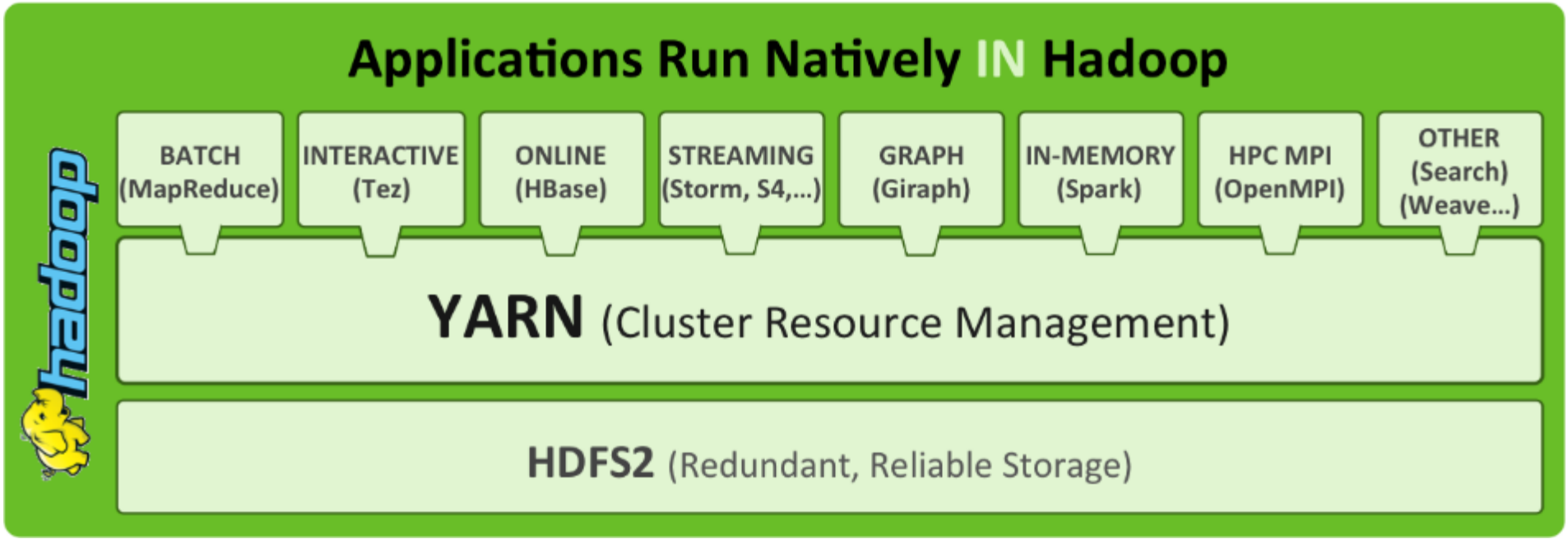
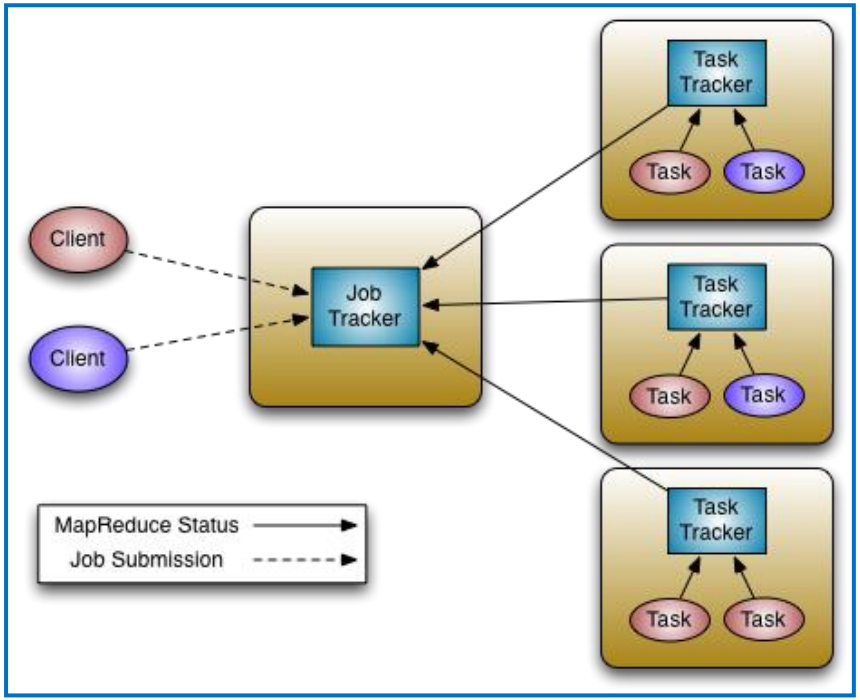


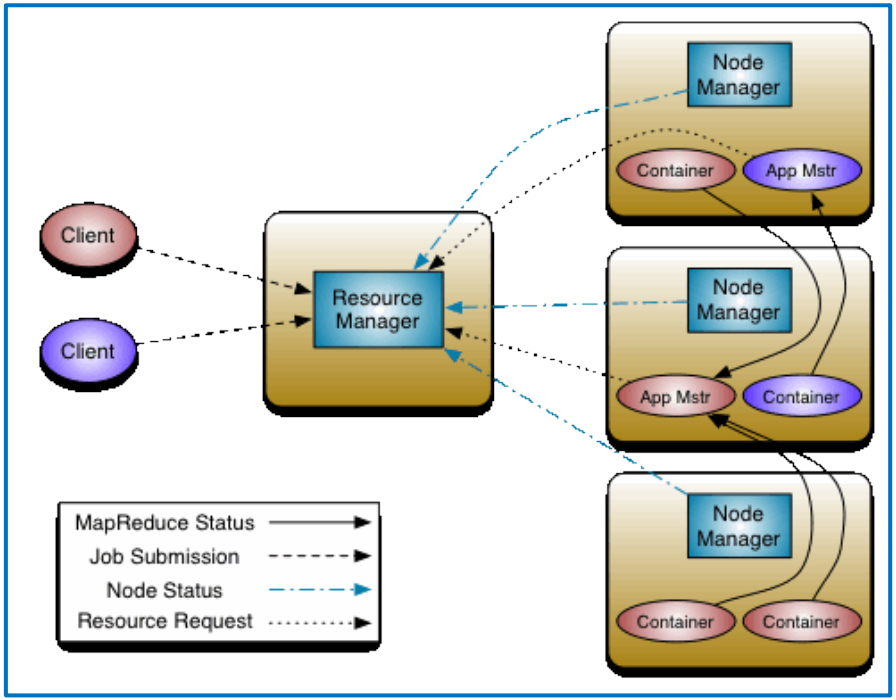
Gráfico tomado de Hortonworks

HADOOP COMPARACION V1 VS. V2 (YARN)

RESOURCE MANAGEMENT MAP-REDUCE



RESOURCE MANAGEMENT YARN



¿QUE ES SPARK?

- Motor de ejecución distribuido “en memoria”.
  - *Map_Reduce necesariamente utiliza discos para pasar los resultados intermedios*
- Modelo de ejecución general que soporta diversos casos de uso, con relativamente baja complejidad de desarrollo.
- Ejecuta en modalidad “stand-alone” o sobre Hadoop.
  - *Se lleva muy bien con Hadoop.*
  - *Compatible con las APIs de almacenamiento de Hadoop.*
  - *Compatible con YARN.*
- APIs nativas en Scala, Java y Python.

SPARK (ON HADOOP) | CONTEO DE PALABRAS

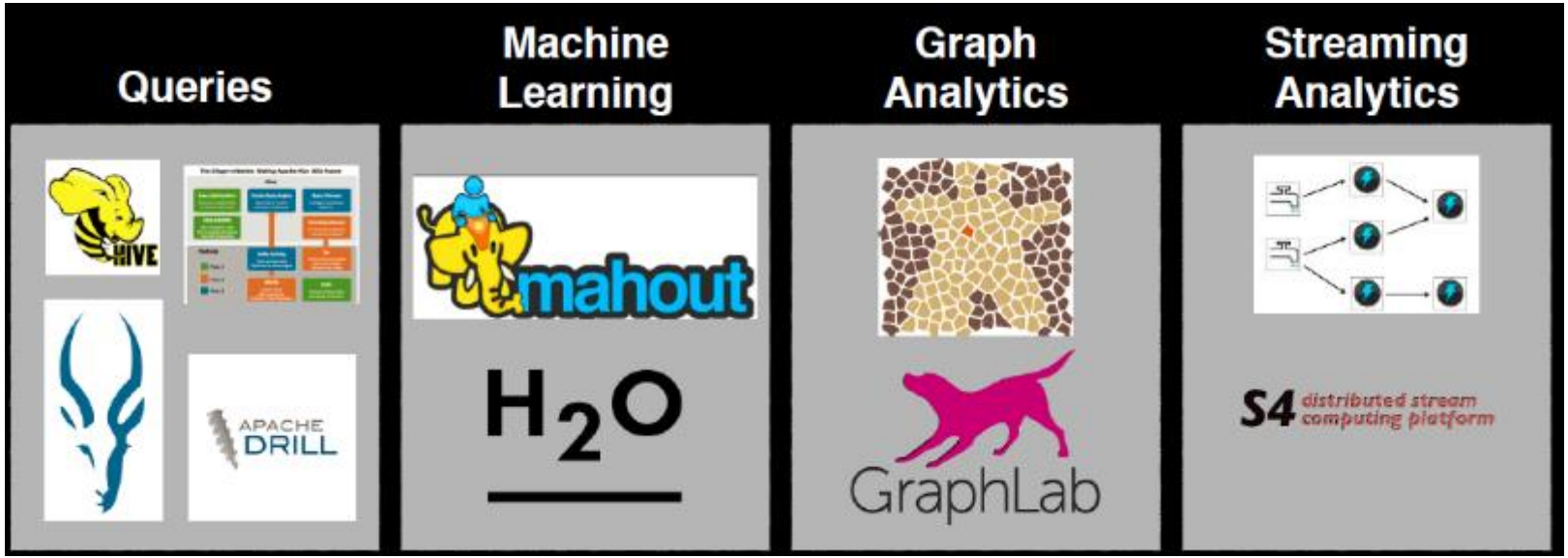
- 50+ líneas en Map_Reduce

```java
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 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);
    }
  }

  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> [<in>...] <out>");
      System.exit(2);
    }
    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);
    for (int i = 0; i < otherArgs.length - 1; ++i) {
      FileInputFormat.addInputPath(job, new Path(otherArgs[i]));
    }
    FileOutputFormat.setOutputPath(job,
      new Path(otherArgs[otherArgs.length - 1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

- 3 líneas en Spark

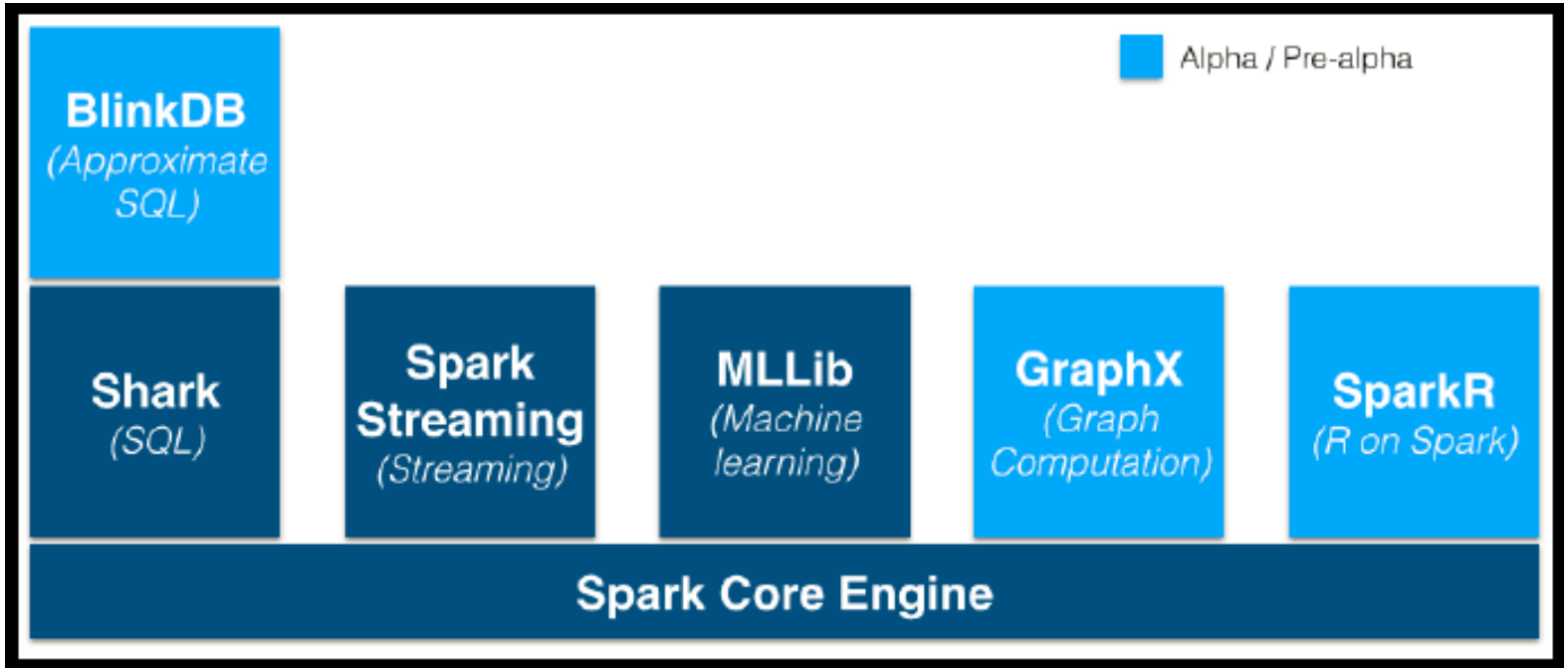
```scala
val f = sc.textFile(inputPath)
val w = f.flatMap(l => l.split(" ")).map(word => (word, 1)).cache()
w.reduceByKey(_ + _).saveAsText(outputPath)
```

- Map-Reduce no se lleva bien con aplicaciones complejas. Entonces comenzaron a aparecer “aplicaciones especializadas”



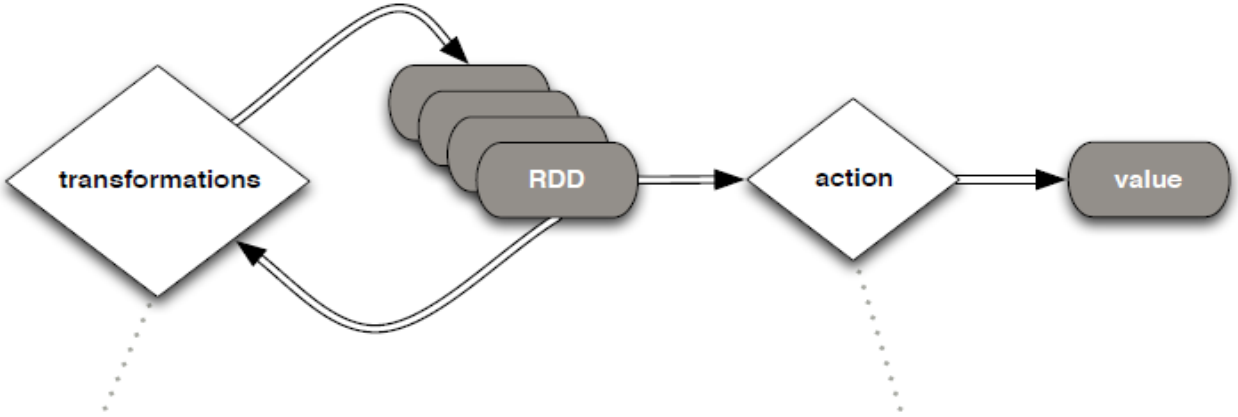
¿QUE ES SPARK?

- Plataforma integrada para analítica sobre Hadoop



SPARK BUILDING BLOCKS

- Resilient Distributed Datasets, Transformaciones y Acciones




```
// transformed RDDs
val errors = lines.filter(_.startsWith("ERROR"))
val messages = errors.map(_.split("\t")).map(r => r(1))
messages.cache()
```

```
// action 1
messages.filter(_.contains("mysql")).count()
```

map, filter, groupBy, sort, union, join, leftOuterJoin, rightOuterJoin, reduce, count, fold, reduceByKey, groupByKey, cogroup, cross, zip, sample, take, first, partitionBy, mapWith, pipe, save

¿40 minutos?

UNIVERSIDAD AUSTRAL

Facultad de Ingeniería

SAS EN HADOOP

sas
THE POWER TO KNOW.

NECESIDAD A RESOLVER

INTERACCION CON SAS CON HADOOP

- Si está planificando o comenzando a utilizar Hadoop
- Si utiliza SAS como plataforma analítica y ahora quiere utilizarla sobre Hadoop.
- Si quiere utilizar Hadoop y no quiere agregar otra herramienta analítica más.
- Si quiere hacer análisis de datos en forma simultánea sobre plataformas RDBMS y Hadoop.
- Si no sabe como analizar los enormes volúmenes de datos que puede almacenar Hadoop.

HOW DOES THE HAND COME? (PARQUE PATRICIOS ENGLISH)
¿COMO VIENE LA MANO?

- Our forecast is that in 2016 SAS will go on being the strongest player on the Hadoop analytical processes arena.
- Their position could be threaten by Apache Spark. SAS will have to constantly improve his analytics offering to avoid be surpassed by this open alternative.

Source: Survey 2014Q4 - KoturSergio Ltd. Inc. Intl. Corp.

NECESIDAD A RESOLVER

ALMACENAR Y ANALIZAR GRANDES VOLUMENES DE INFORMACION A BAJO COSTO

TODAS LAS MODALIDADES

- Facilitando la implementación y ejecución de todas las modalidades.
- En forma progresiva o consolidada.

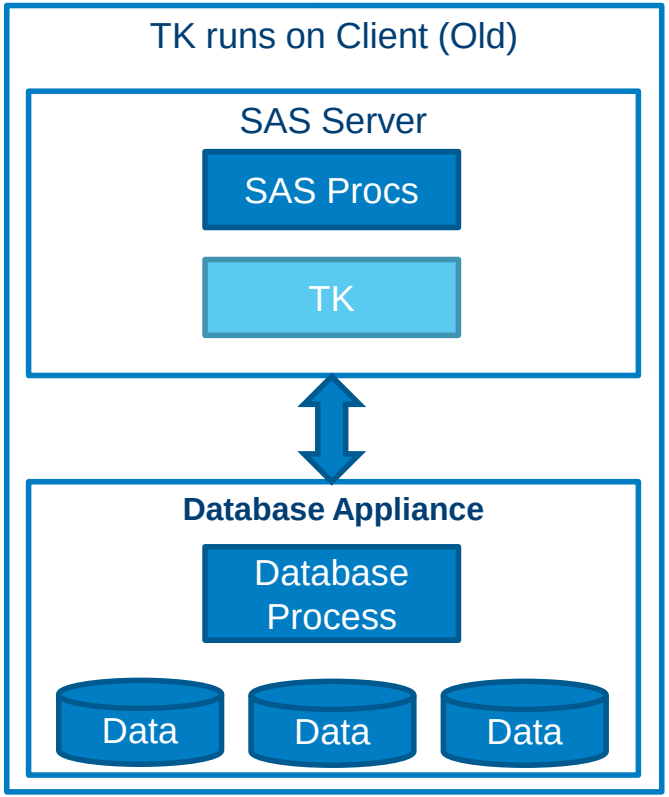
COMO REPOSITORIO DE INFORMACION

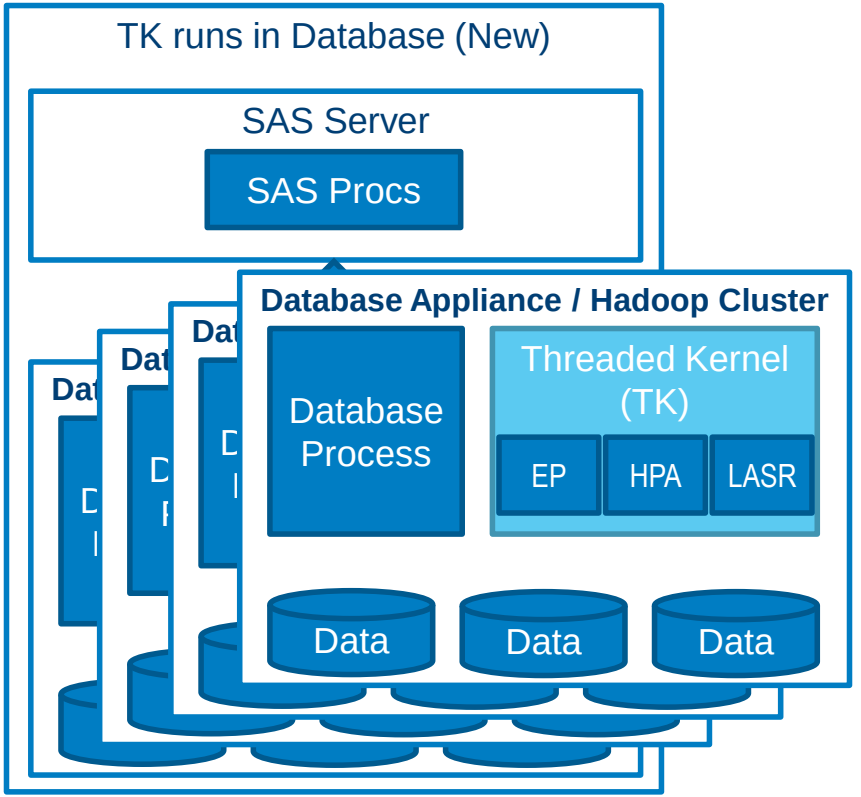
+ PROCESAMIENTO EN PARALELO MODALIDAD MAP-REDUCE

+ PROCESAMIENTO EN PARALELO MODALIDAD IN-MEMORY

SAS PROCESAMIENTO EN PARALELO

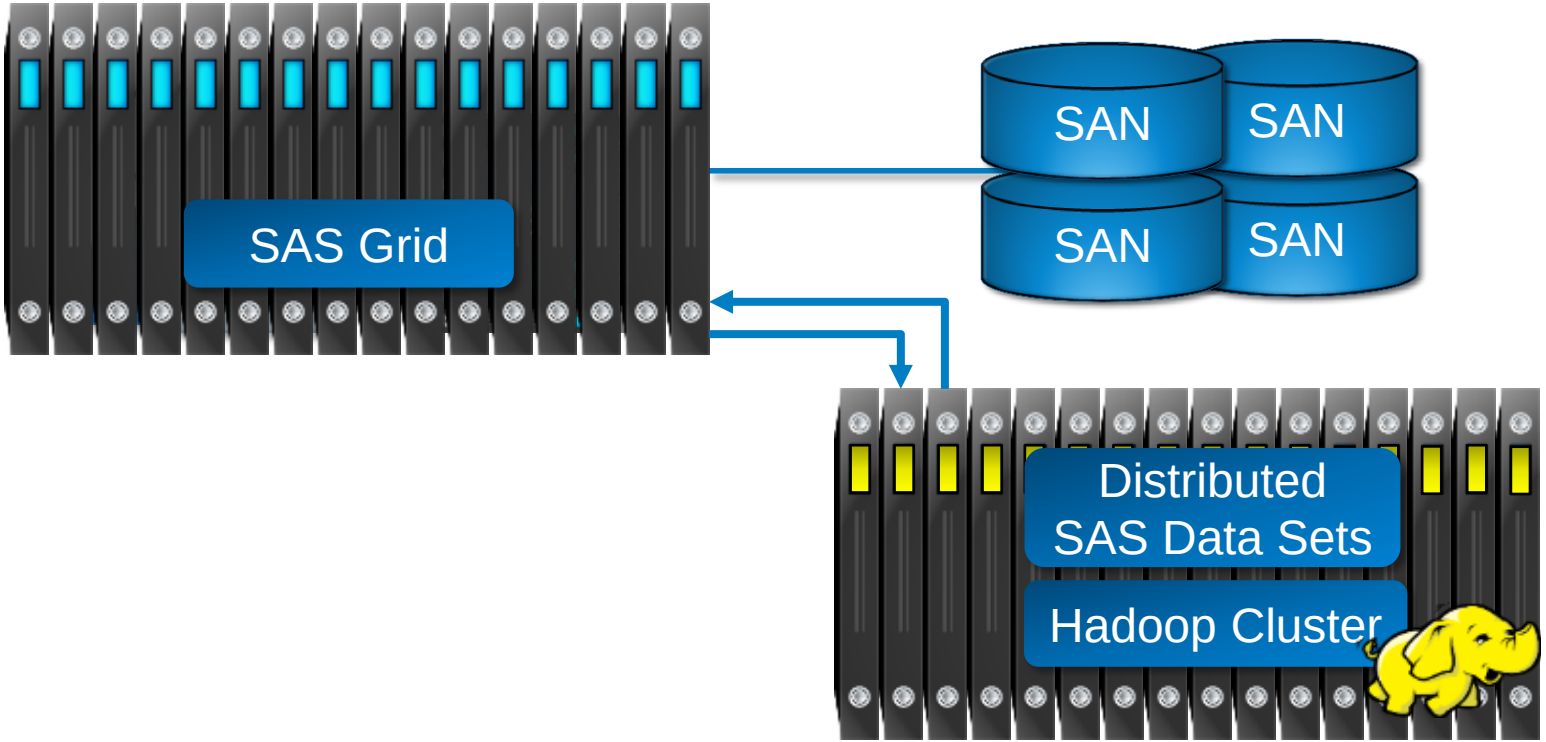
EJECUCION DE THREADED KERNEL EN LOS APPLIANCES DE BASES DE DATOS Y CLUSTERS HADOOP



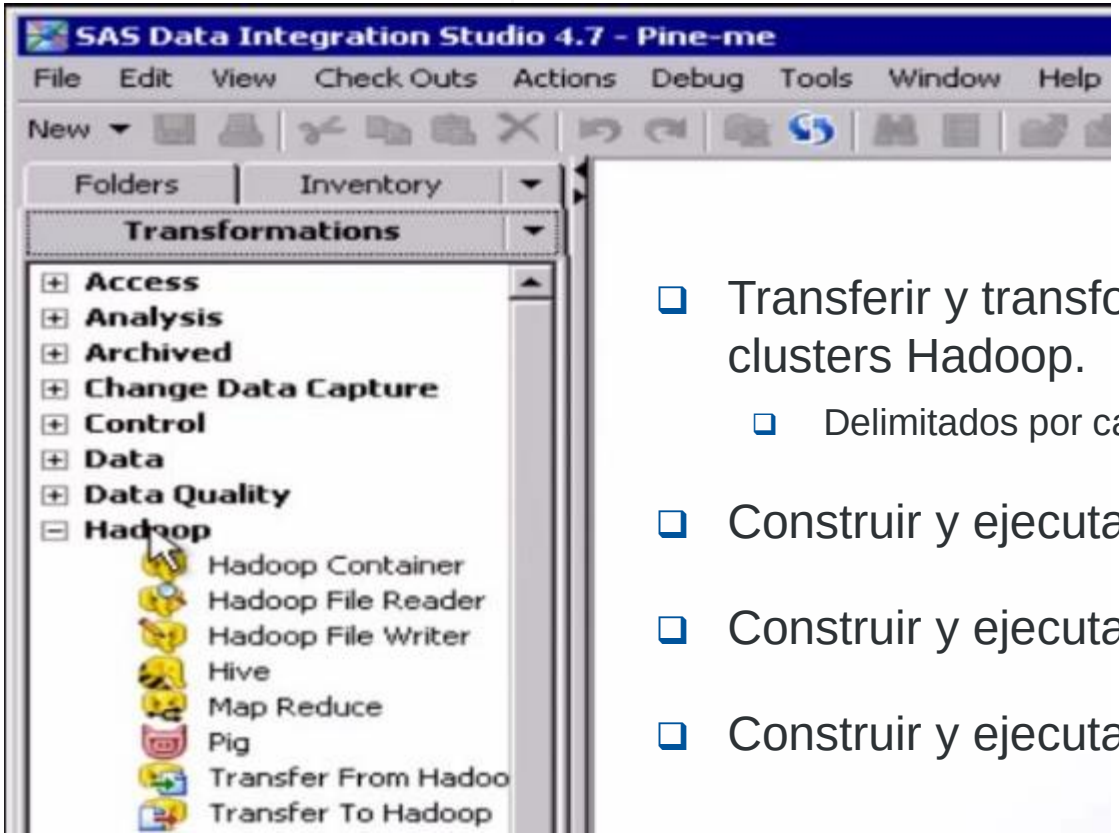


SAS SOBRE HADOOP COMO REPOSITORIO DE DATOS ANALITICOS

HADOOP COMO REPOSITORIO DE DATOS ANALÍTICOS



HADOOP COMO REPOSITORIO DE DATOS ANALÍTICOS
OPERACIONES PRIMARIAS CON SAS DATA INTEGRATION



- Transferir y transformar tablas SAS desde y hacia clusters Hadoop.
  - Delimitados por caracteres, XML, JSON, entre otros
- Construir y ejecutar programas Map-Reduce.
- Construir y ejecutar programas Pig.
- Construir y ejecutar programas Hive.

OPERACIONES HADOOP DESDE PROGRAMAS SAS PIG Y MAP-REDUCE

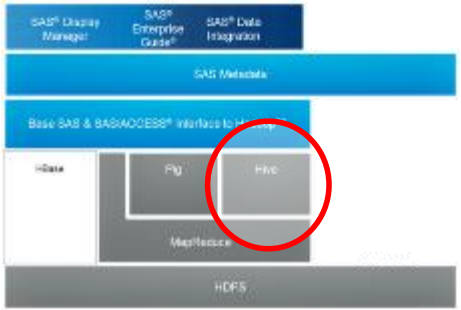

```
/*
Run PIG script
*/
filename cfg "C:\Sample_Data\hadoop_config.xml";
filename pigcode1 "C:\Sample_Data\pig_cd.txt";
proc hadoop options=cfg username="hadoop" password="hadoop"
verbose;
    pig code=pigcode1 ;
run;
```

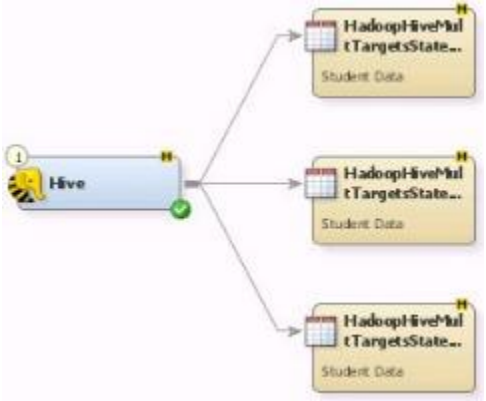
El sgte. script PIG Latin script lee el archivo *NYSE_dividendos* desde HDFS y lo agrupa por 'simbolo'. Luego, calcula el promedio de dividendos de cada grupo y lo almacena en la carpeta HDFS *d_promedio_dividendos*.


```
CD NYSE ;
d_dividendos = LOAD 'NYSE_dividendos' as (d_exchange, d_simbolo, d_fecha, d_dividendo_ind);
d_grouped    = GROUP d_dividendos BY d_simbolo;
d_promedio   =  FOREACH d_grouped  GENERATE group,  AVG(d_dividendos.d_dividendo_ind);
STORE d_promedio INTO 'd_promedio_dividendos';
```

SAS SOBRE HADOOP

OPERACIONES HADOOP COMO CON CUALQUIER DB HIVE






```
LIBNAME cdh_hdp HADOOP  PORT=10000 SERVER=sascldserv02 user=hadoop password=hadoop ;
/*
Create new table
*/
proc sql;
connect to hadoop(PORT=10000 SERVER=sascldserv02  USER=hadoop  PASSWORD="hadoop");
exec( create table cars_prc (make string,  model string,  msrp double) ) by hadoop;
quit;

/*
Copy from another table
*/
proc sql;
  insert into cdh_hdp.cars_prc
  select make, model, msrp
  from sashelp.cars ;
quit;
/*
List contents
*/
proc sql;
select * from cdh_hdp.cars_prc;
quit;
```

Beneficios

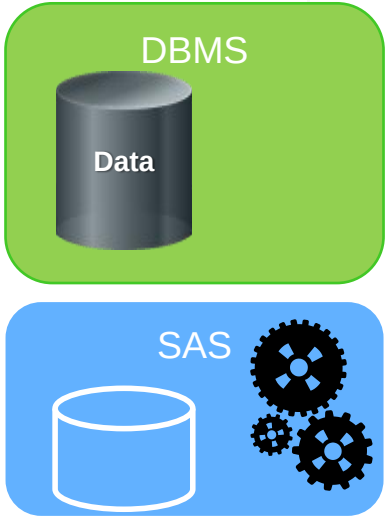
- Sustanciales ahorros de costos de almacenamiento
- Mejora de performance
- Sin límite en el ancho de las tablas soportadas (>2000 variables)
- Optimización del costo de protección de datos mediante la replicación HDFS

SAS SOBRE HADOOP IN-DATABASE

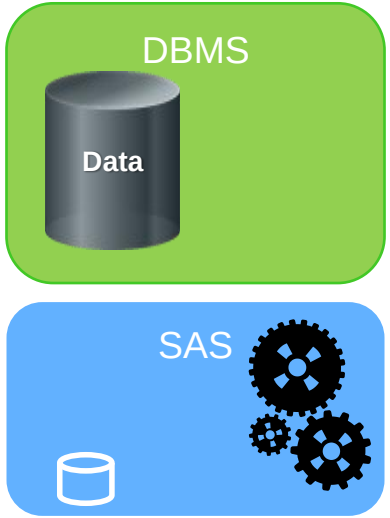
SAS IN-DATABASE

HADOOP COMO REPOSITORIO Y PARA PROCESAMIENTO MODALIDAD MAP-REDUCE

MODALIDAD SAS NATIVA



MODALIDAD SAS IN-DATABASE



Modo tradicional de operación SAS

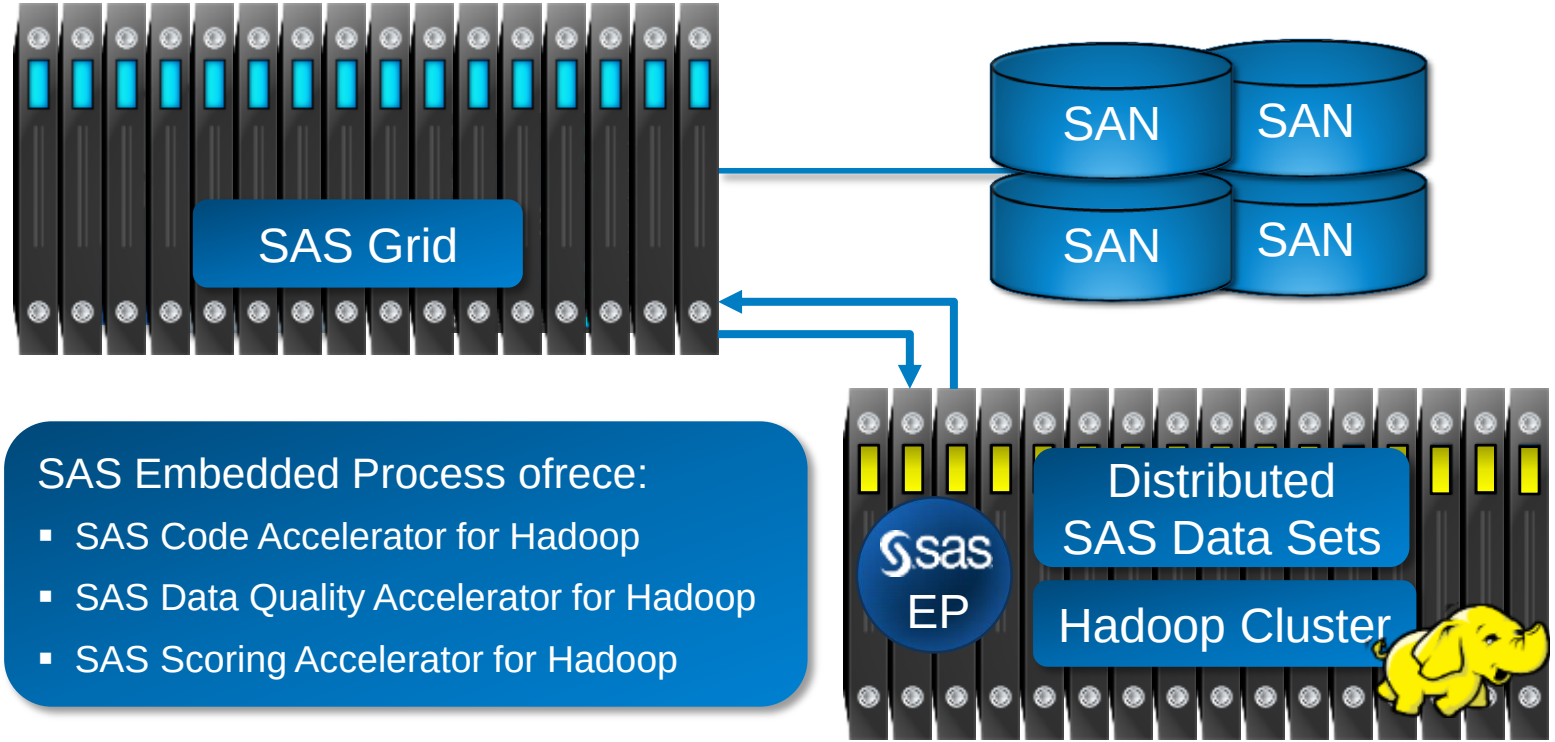
- Actividades que requieren entorno de ejecución SAS
  - Que no pueden ser ejecutadas mediante SQL
  - Que ejecutan mejor en entornos nativos SAS
  - Soluciones SAS

SAS In-Database

- Transformaciones de SAS Data Integration
- Reportes basados en queries SQL
- Lenguaje DS2
- SAS Scoring Accelerator
- SAS Data Quality Accelerator

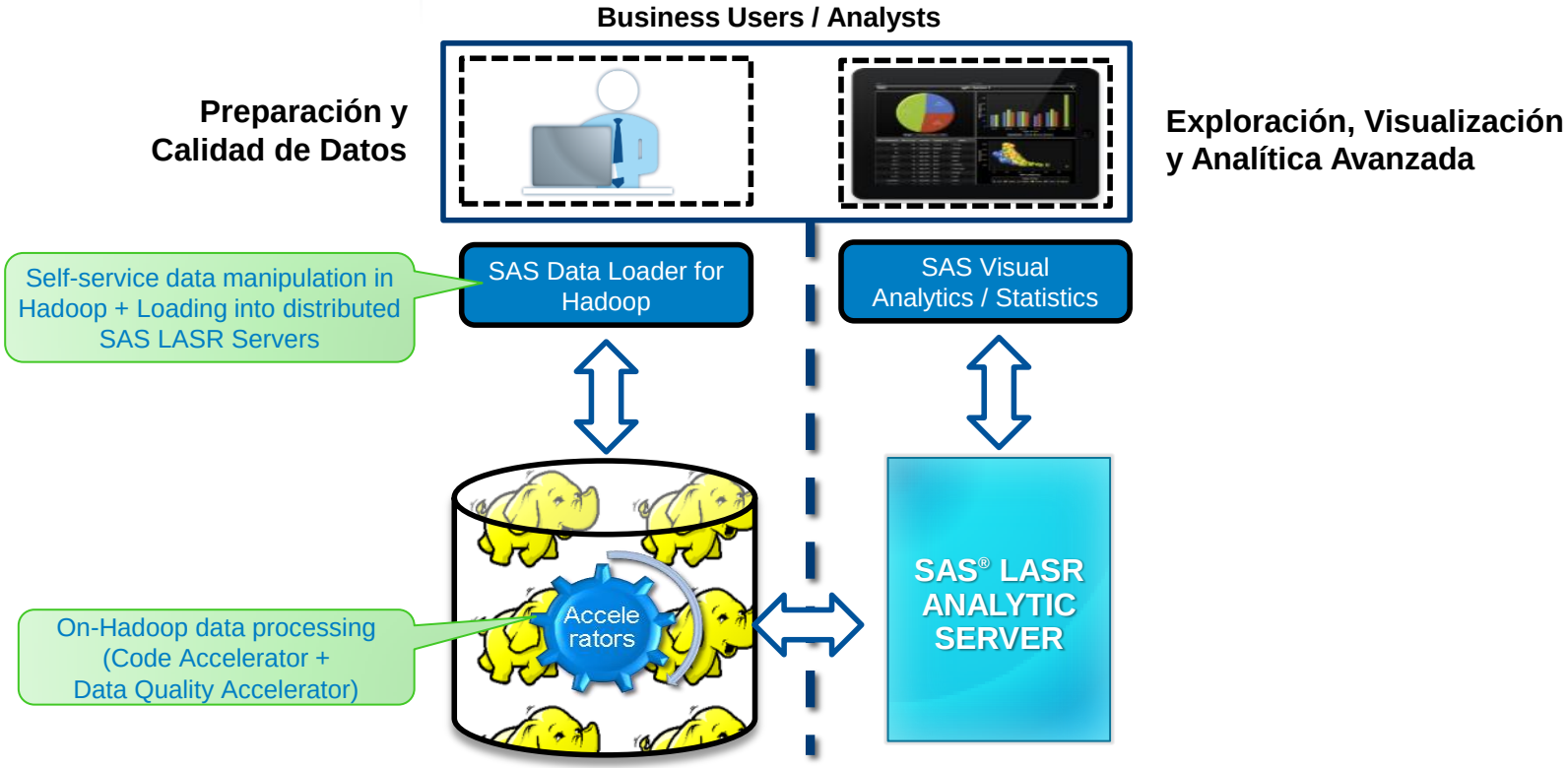
SAS IN-DATABASE SOBRE HADOOP

HADOOP COMO REPOSITORIO Y PARA PROCESAMIENTO MODALIDAD MAP-REDUCE



FACILITANDO HADOOP

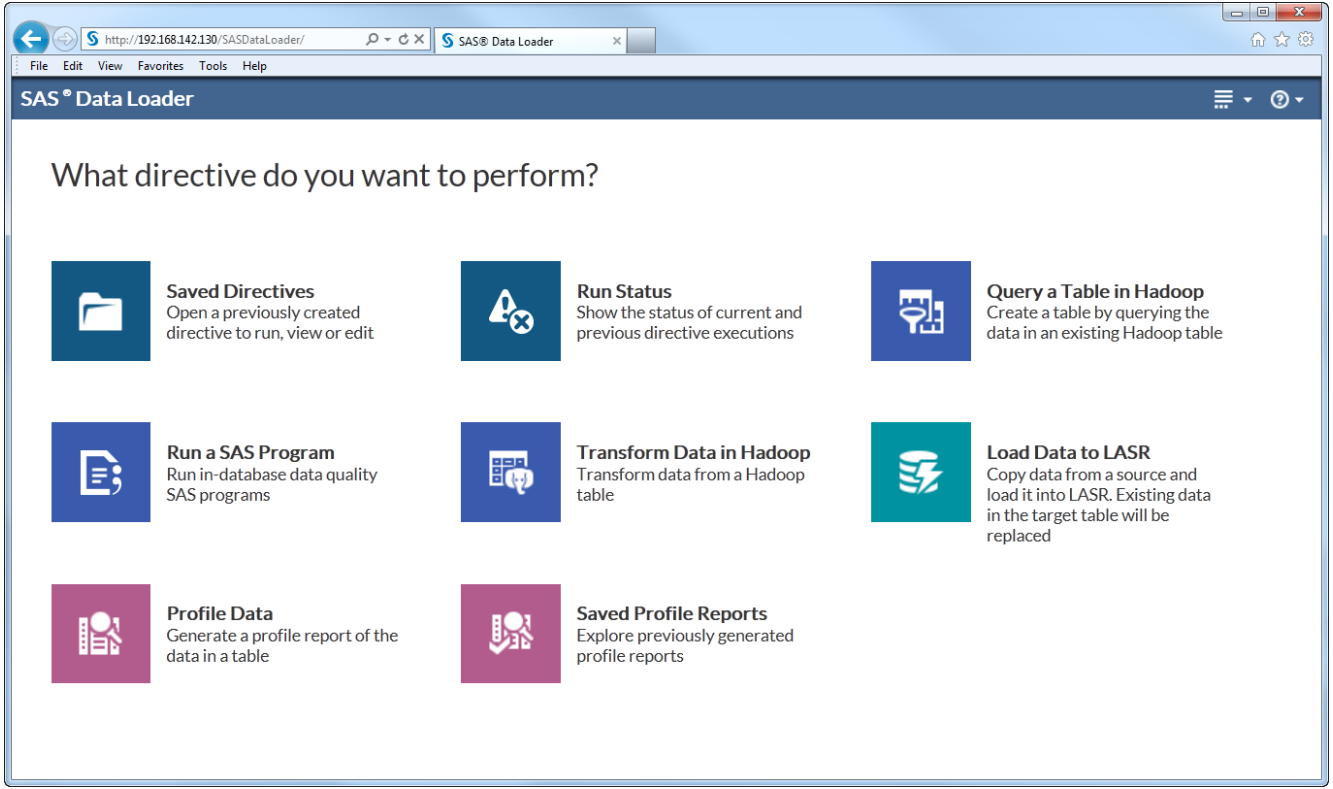
SAS DATA LOADER FOR HADOOP



SAS DATA LOADER FOR HADOOP 2.1

- Autoservicio de datos para los usuarios
- Sin necesidad de codificación o scripting
- Sin necesidad de conocimientos especializados en Hadoop

MENU PRINCIPAL



Beneficios

- Grandes mejoras de performance
- Gran crecimiento del poder de cómputo basado en el procesamiento en paralelo de Hadoop
- Significativa reducción del movimiento de datos

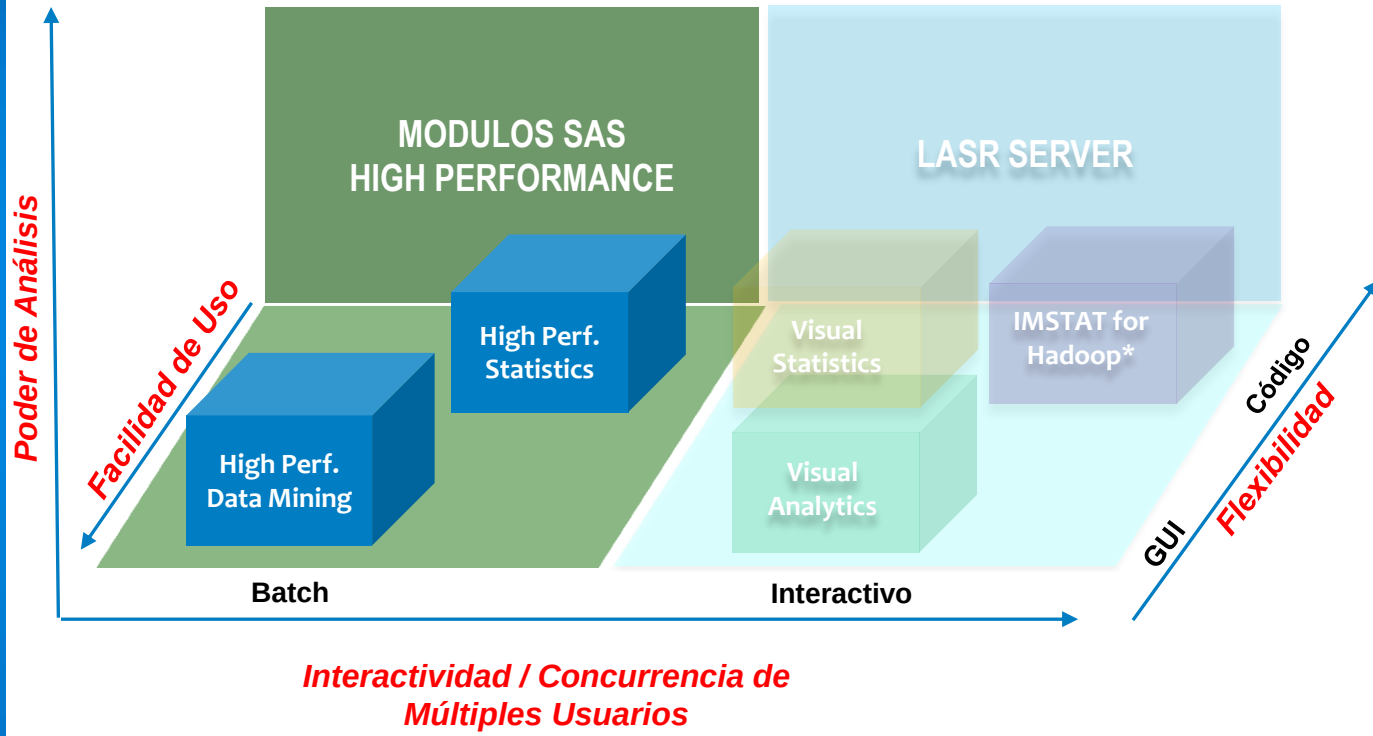
SAS SOBRE HADOOP
IN-MEMORY
HIGH-PERFORMANCE ANALYTICS

SOLUCIONES ANALÍTICAS

PROCESAMIENTO EN PARALELO EN MEMORIA

DIFERENCIADORES DE LOS PRODUCTOS SAS:

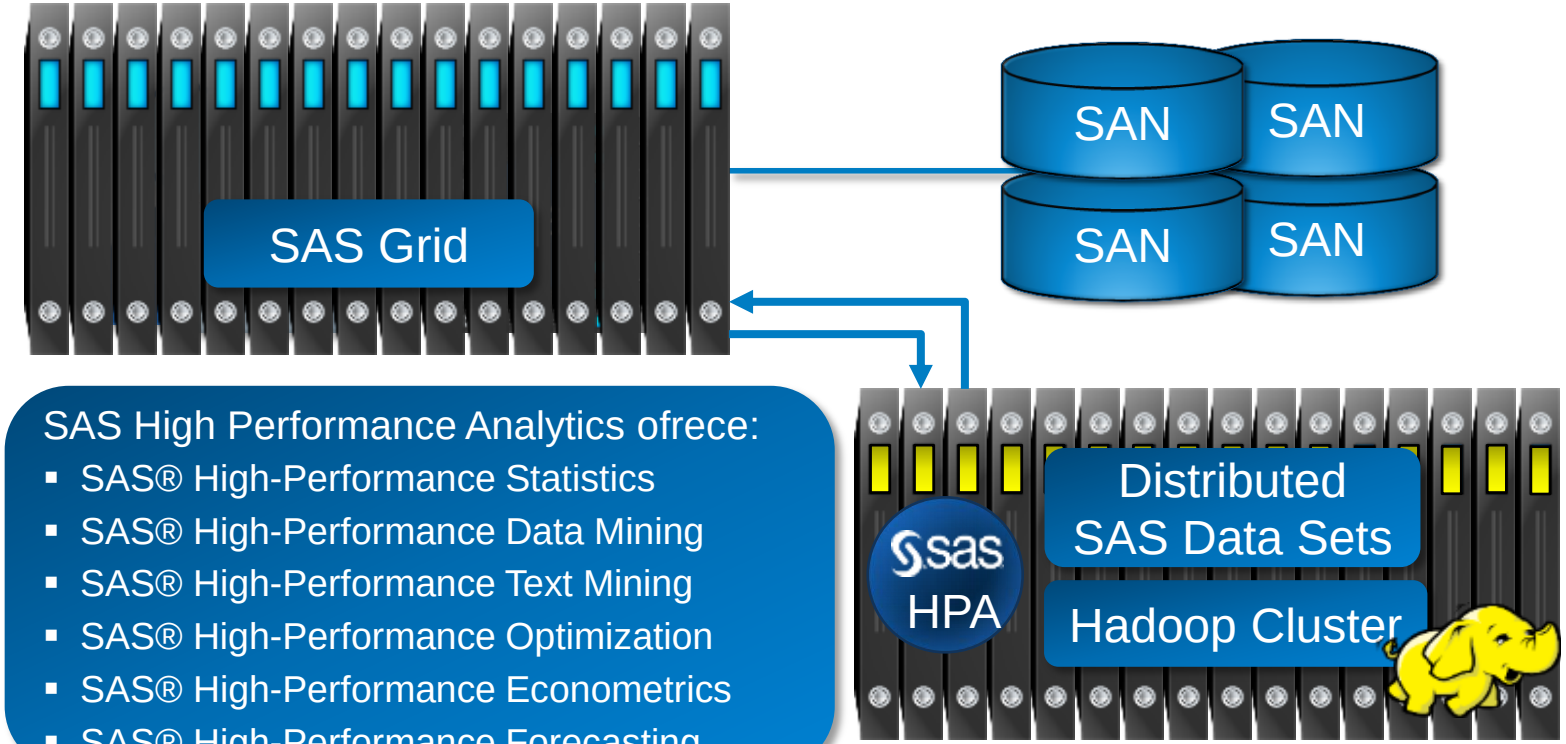
- PODER DE ANÁLISIS
- INTERCATIVIDAD / CONCURRENCIA
- DE MÚLTIPLES USUARIOS
- FLEXIBILIDAD / FACILIDAD DE USO



*SAS® In-Memory Statistics for Hadoop

SAS SOBRE HADOOP

METODOS SAS NATIVOS AHORA EJECUTANDO EN MODALIDAD IN-MEMORY

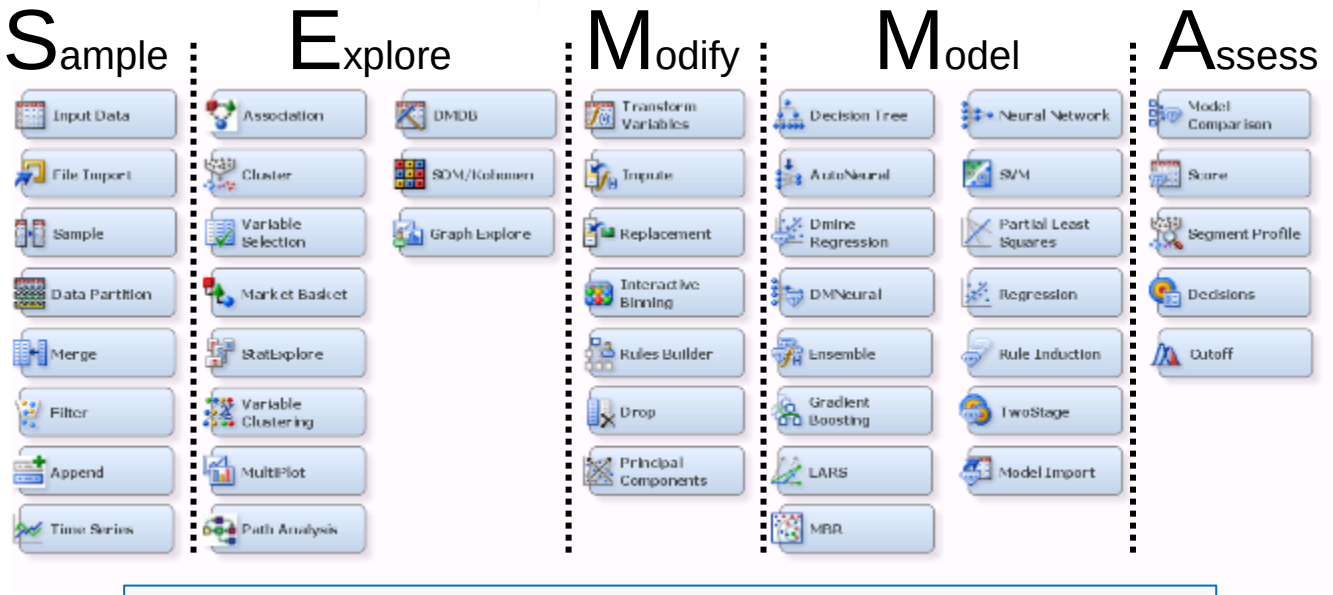


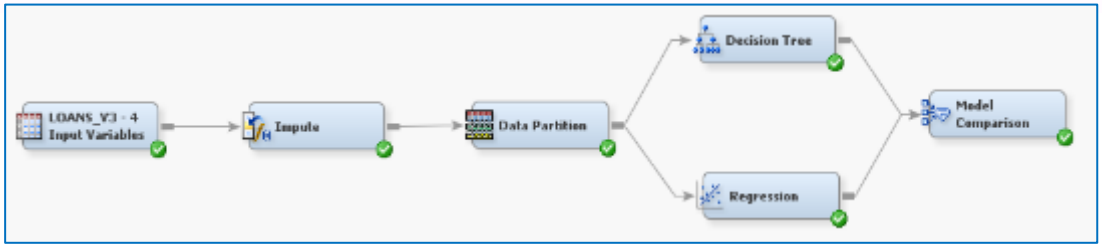
SAS High Performance Analytics ofrece:

- SAS® High-Performance Statistics
- SAS® High-Performance Data Mining
- SAS® High-Performance Text Mining
- SAS® High-Performance Optimization
- SAS® High-Performance Econometrics
- SAS® High-Performance Forecasting

SAS SOBRE HADOOP

METODOS SAS NATIVOS
AHORA EJECUTANDO EN MODALIDAD IN-MEMORY





CASO DE USO

Modelo de propénsión a compra en cadena de hoteles;
Regresión logística;
20 horas vs 20 minutos;
a igual inversión

SAS SOBRE HADOOP
IN-MEMORY

- VISUAL ANALYTICS
- VISUAL STATISTICS
- IN-MEMORY STATISTICS FOR HADOOP

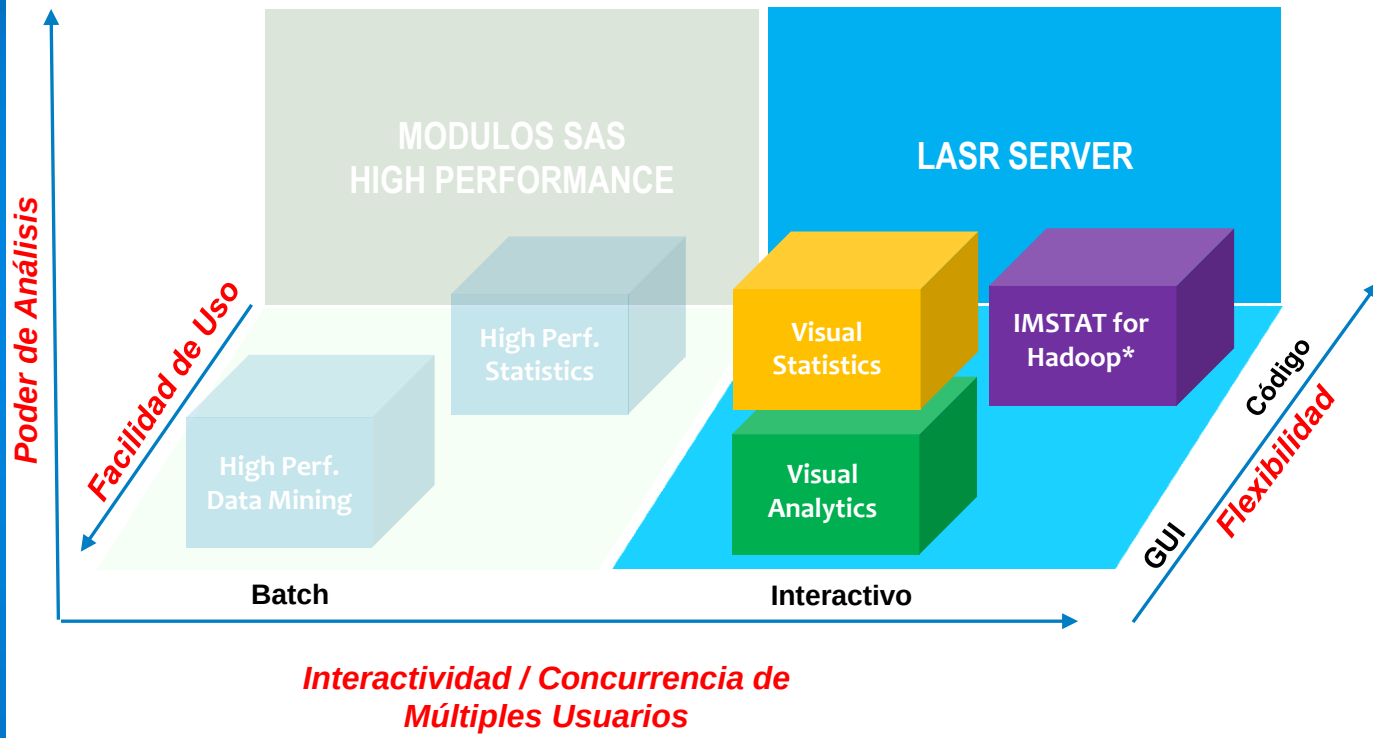
...TODOS BASADOS EN LASR SERVER

SOLUCIONES ANALÍTICAS

PROCESAMIENTO EN PARALELO EN MEMORIA

DIFERENCIADORES DE LOS PRODUCTOS SAS:

- PODER DE ANÁLISIS
- INTERCATIVIDAD / CONCURRENCIA
- DE MÚLTIPLES USUARIOS
- FLEXIBILIDAD / FACILIDAD DE USO



*SAS® In-Memory Statistics for Hadoop

SERVIDOR ANALÍTICO... ¿QUÉ ES UN SERVIDOR ANALÍTICO?

LASR ANALYTICS SERVER

SERVIDOR ANALÍTICO: SUS INSTRUCCIONES EJECUTAN DIRECTAMENTE FUNCIONES ANALITICAS (ACCIONES)

SOBRE TABLAS EN MEMORIA: SASIOLA ENGINE; EN MODALIDAD STATELESS

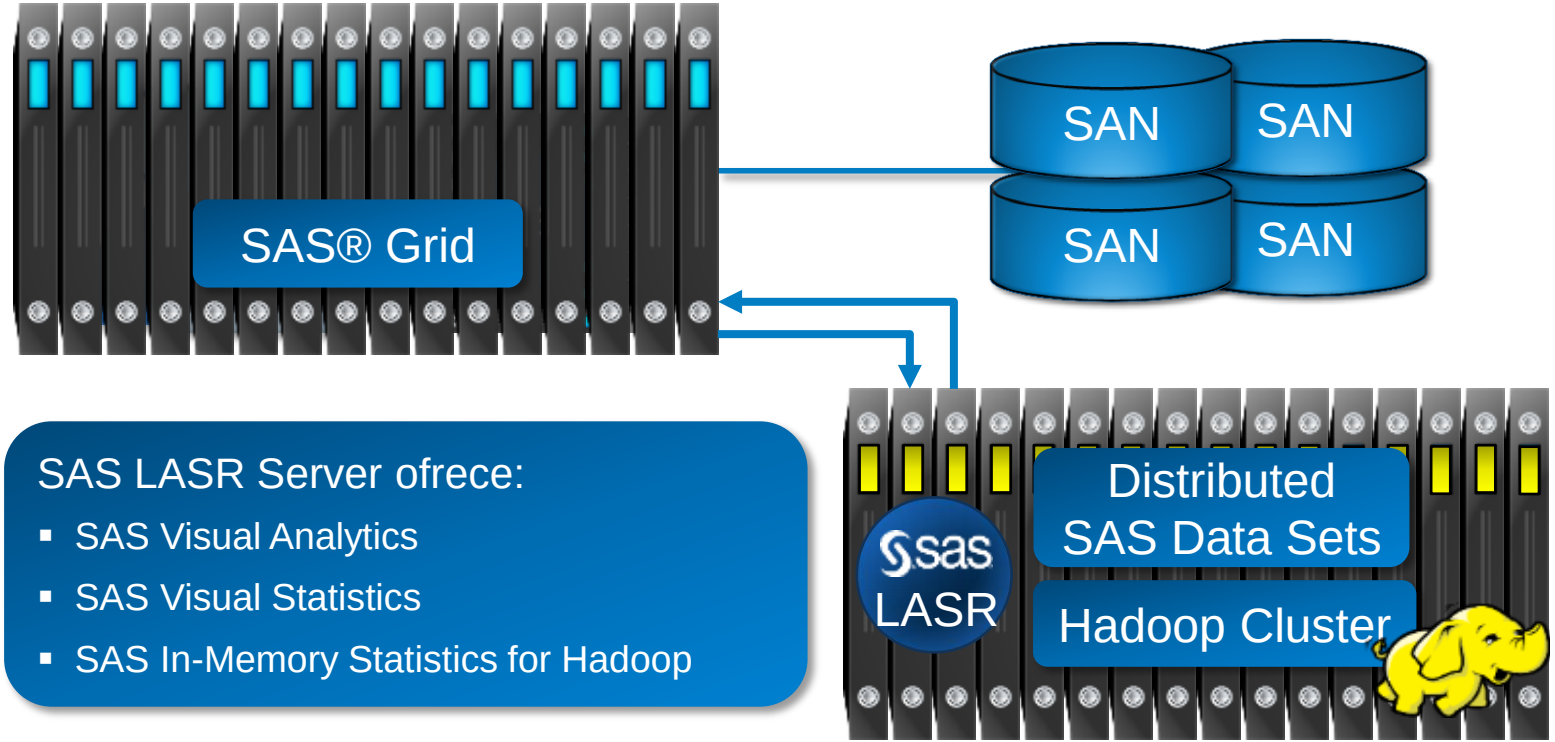
NOMINA DE ACCIONES

ADDTABLE, ADDTKHPSTABLE, APPENDTABLE, ASSESS, BOXPLOT, CLASSLEVELS, CLUSTER, COLUMNINFO, CONTOURPLOT, COMPUTEDCOLUMN, CORRELATION, CROSSTAB, DECISIONTREE, DELETEROWS, DIRECTLOAD, DISTINCTCOUNT, DISTRIBUTIONINFO, DROP, DROPTABLE, EXPORT, EXTERNAL, FETCHROWS, FITMODEL, FORECASTSERIES, FREQUENCY, GROUPBY, HISTOGRAM, IMPORT, IMPORTCUBE, KERNELDENS, LIFETIME, LISTSORTS, MDSUMMARY, NUMROWS, OPTIMIZE, PARALLELCOORDINATES, PARTITION, PARTITIONINFO, PERCENTILE, PROMOTE, PSPLINE, PURGETEMPTABLES, RANDOMFOREST, REALSCATTER, RECOMMEND, REGCORR, REGRESSION, SAVETABLE, SCHEMA, SCORE, SERVERINFO, SERVERPARM, SERVERVERSION, SETTABLES, SCATTERPLOT, SCATTERPLOTMATRIX, SORTORDER, SUMMARY, TABLEINFO, TERMINATE, TEXTPARSE, TOPK, UPDATE

- INTERFAZ CLIENTE JAVA: VISUAL ANALYTICS, VISUAL STATISTICS
- INTERFAZ IMSTAT: PROGRAMACION SAS
- INTERFACES PARTICULARES: VASMP, HPAML
- INTERFAZ PROGRAMACION C

SAS LASR ANALYTICS SERVER

HADOOP COMO REPOSITORIO Y PARA PROCESAMIENTO MODALIDAD IN-MEMORY

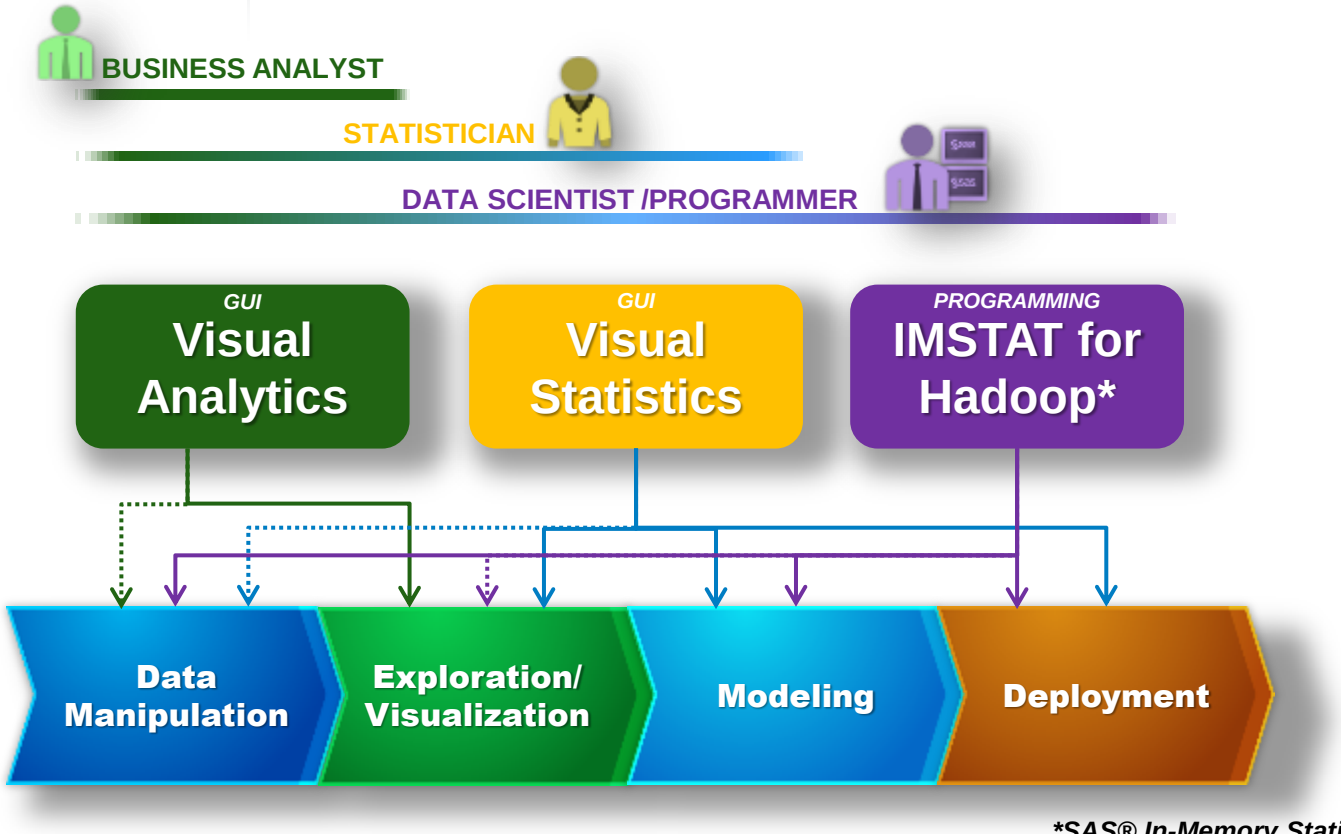


SAS LASR Server ofrece:

- SAS Visual Analytics
- SAS Visual Statistics
- SAS In-Memory Statistics for Hadoop

SAS LASR ANALYTIC SERVER

INTERFACES PARA LAS DIFERENTES NECESIDADES DE LOS DIFERENTES TIPOS DE USUARIOS



**SAS® In-Memory Statistics for Hadoop*

SAS VISUAL ANALYTICS

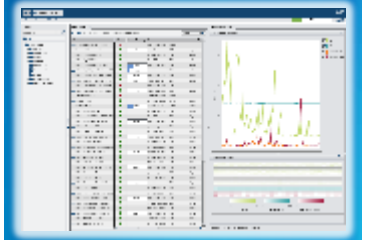
ANALISIS EXPLORATORIOS ESTADÍSTICA Y MINERÍA DE DATOS BÁSICAS

UNIVERSIDAD AUSTRAL
Facultad de Ingeniería

Central Entry Point | Integration | Role-based Views

DATA BUILDER

- Operaciones relacionales desde diversas fuentes
- Creación de columnas calculadas
- Carga de datos



ADMINISTRATOR

- Monitor SAS® LASR™ Analytic server
- Carga y descarga de datos
- Administración de seguridad

EXPLORER

- Descubrimiento de patrones y análisis de datos
- Analítica avanzada

DESIGNER

- Creación de reports del tipo tablero de control para visualizarlos en la Web o en dispositivos móviles

MOBILE BI

- Aplicaciones para iOS y Android para visualización interactive de reportes

SAS® LASR™ ANALYTIC SERVER

sas THE POWER TO KNOW.

SAS VISUAL ANALYTICS

VERSION 7.1

- **Reporting Enhancements**
  - Printing
  - Filters on Aggregates
  - Global Prompts
  - Parameterised Calculations / Display Rules / Filters / Ranks
  - New Prompt Container (*real-estate utilization)*
  - Scheduling / Distribution
  - Graphical Enhancements *(Word Cloud, Graph Gallery, Animations, 100% Stacked Bar Chart)*
  - Alerting Enhancements
  - More statistics (*aggregation options)*
  - % based Ranks
  - Interactive Pop-up Visuals *(real-estate utilization)*
- **Administration Enhancements**
  - Refined memory mapping information
  - Audit Reporting
- **SAS Home and Viewer in HTML5***
- **Updated SAS Mobile BI Interface**
- **Data Management Enhancements**
  - Support for xlsb and xlsm files
  - Support for Pivotal Hadoop / Impala
  - Ability push down more to database
  - Compression
- **Exploration Enhancements**
  - Support for multiple data sources
  - Parameterised Calculations
  - Advanced filter interactivity (*support for Train of Thought Analysis)*
  - Goal-Seeking
  - Word Cloud with Sentiments Analysis (*with some modification options)*
  - More statistics (*aggregation options)*
  - Path Analysis visualization
- **Office Analytics Enhancements**
  - EG → New task: Upload to LASR
  - Outlook → Improved performance on content rendering
  - Interactivity Window for Office tools

Entorno totalmente interactive para el modelado estadístico en forma visual

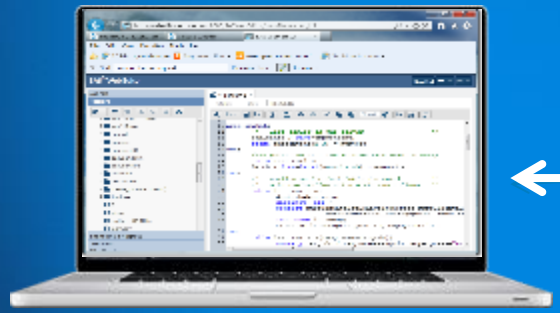
- Los modelos estadísticos ejecutan en SAS LASR Server
- Misma interfaz de usuario que Visual Analytics

Funcionalidades principales

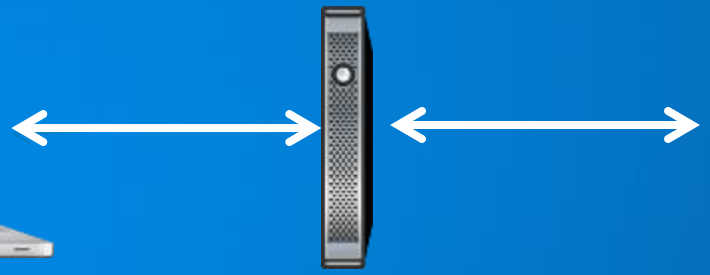
- Múltiples usuarios concurrentes sobre copia única de los datos.
- Estadística clásica: Regresiones multiples, Regresión logística, Análisis de varianza, Modelo lineal generalizado, Clustering.
- Estadística moderna / Machine learning (Árboles de decisión, Random forest, Clasificadores de Bayes ingenuos).
- Procesamiento GROUP BY en paralelo.
- Descubrimiento visual de puntos críticos como outliers y puntos de influencia.

SAS IN-MEMORY STATISTICS FOR HADOOP

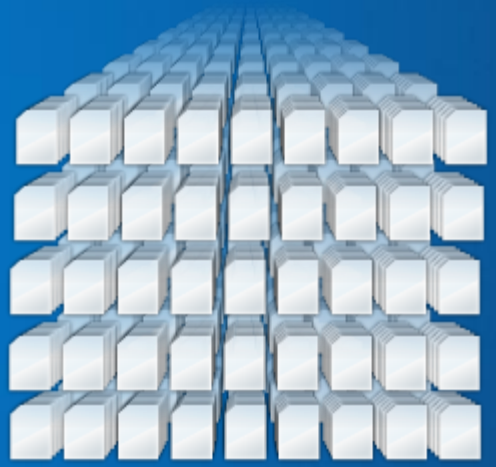
ACCESO PROGRAMÁTICO A TODA LA FUNCIONALIDAD DEL LASR SERVER



SAS WebOne HTML 5 Modern Coding Environment
~ Happy Data Scientists & SAS Analysts



SAS Server
~ BASE, ODS, Access to Hadoop ,LASR
12 bytes (IMSTAT, RECOMMEND Etc..)



LASR Analytic Server on Hadoop

SAS IN-MEMORY STATISTICS FOR HADOOP

Data Manipulation
- SAS Data Step
- BALANCE
- COLUMINFO
- COMPUTE
- DELETEROWS
- DISTINCT
- DROPTABLE
- FETCH
- GROUPBY
- PARTITION
- PROMOTE
- PURGETEMPTABLES
- SET
- TABLE
- UPDATE

Data Exploration/ Visualization
- BOXPLOT
- CORR
- CROSSTAB
- CONTOURPLOT
- DISTRIBUTIONINFO
- FREQUENCY
- HISTOGRAM
- KDE
- REPLAY
- SUMMARY

Predictive Modeling
- DECISIONTREE
- FORECAST
- GENMODEL
- GLM
- RANDOMWOODS
- ASSESSMENT

Descriptive Modeling
- CLUSTER
- CLUSTER TF-IDF
- ASSOCIATIONS
- SVD

Recommender
- CLUSTER
- KNN
- ASSOCIATIONS
- SVD

Text Analytics
- PARSING
- SVD

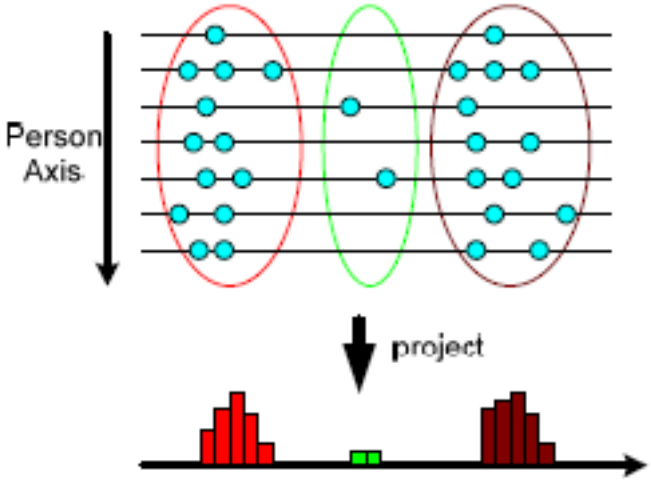
Miscellaneous
- EXTERNAL (C API)
- FREE
- SAVE
- STORE

Deployment
- SCORE

Beneficios

- Incremento masivo de la performance
- Simplificación de la administración de recursos
- Optimización del movimiento de datos mediante gran paralelismo
- Adaptado para desarrollar / ejecutar todo tipo de modelos analíticos

Caso Modelo Fraude Alta Complejidad

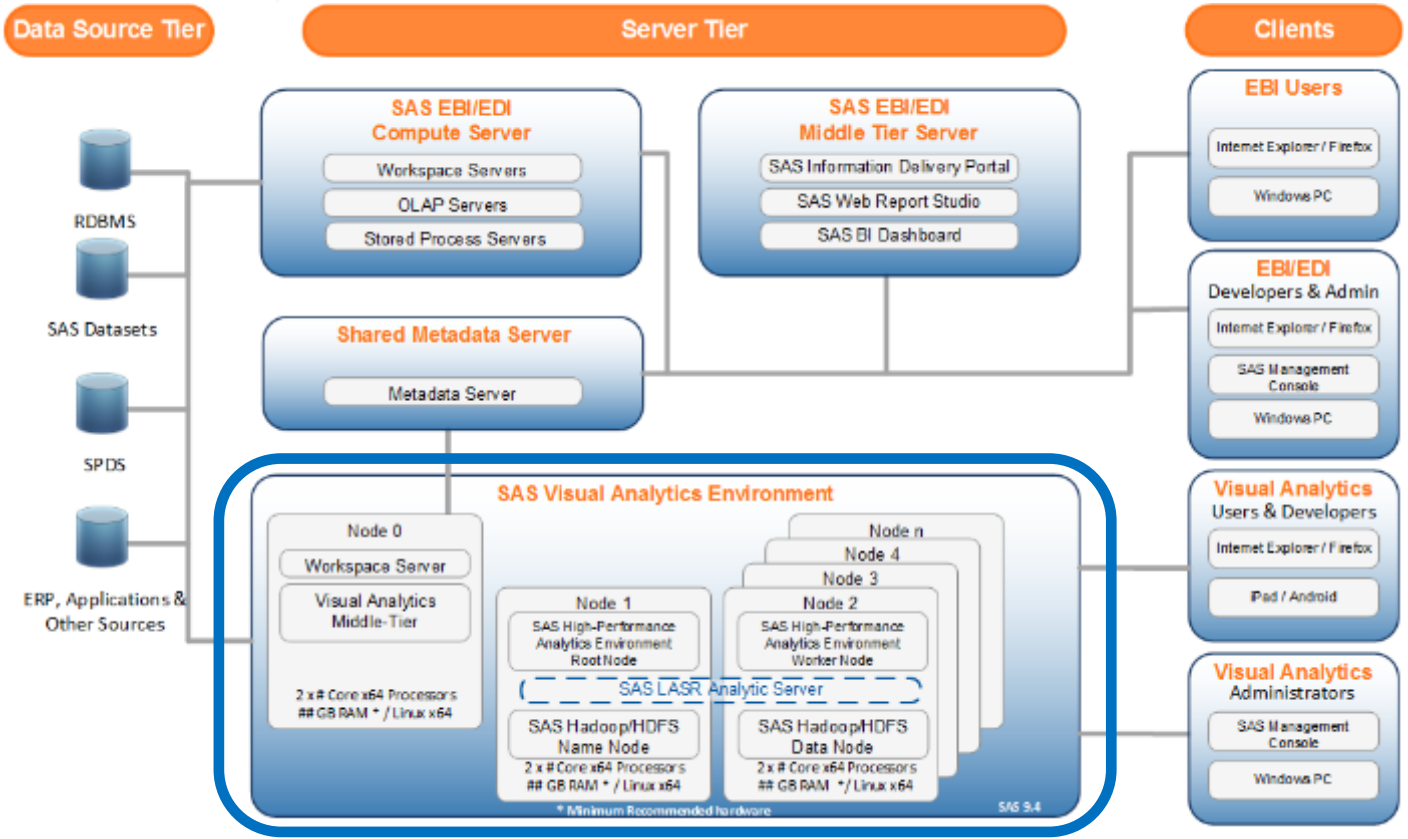


Clustering + Impacto + Tiempo + Secuencia

LINEAMIENTOS PARA LA IMPLEMENTACION DE UN LABORATORIO ANALITICO DE SAS SOBRE HADOOP

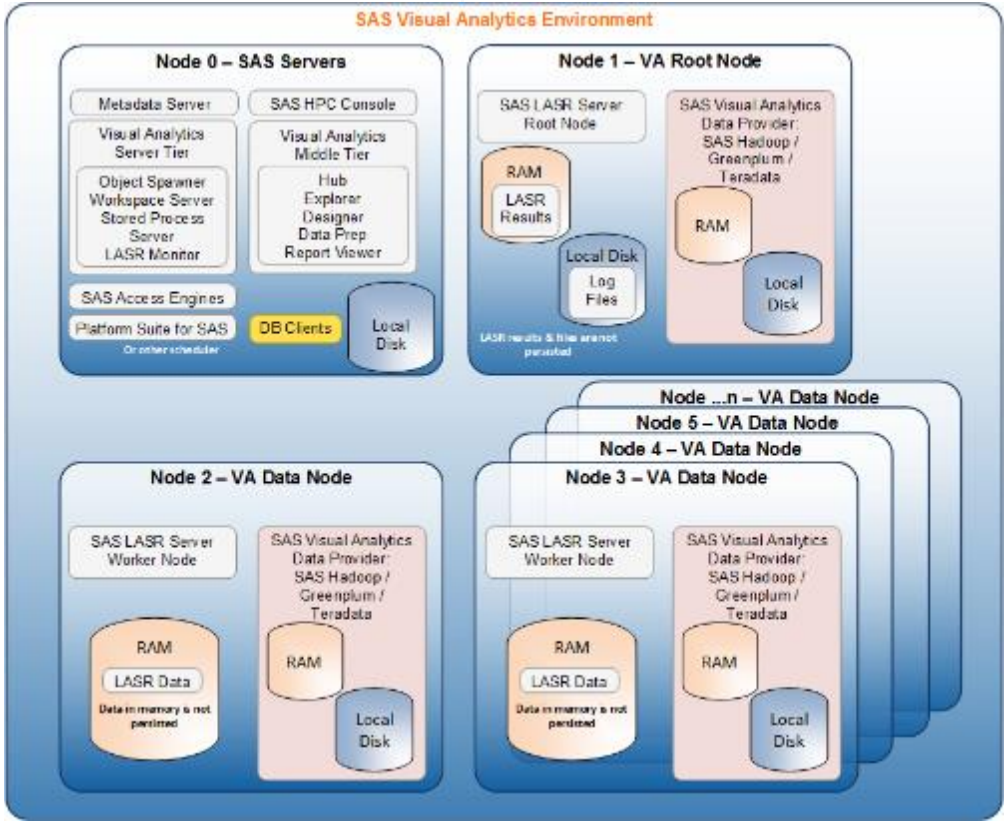
SAS LASR ANALYTIC SERVER

RELACION LASR SERVER CON LA ARQUITECTURA GENERAL DE SAS



SAS LASR ANALYTIC SERVER

DENTRO DE LOS SERVIDORES DEL LASR SERVER

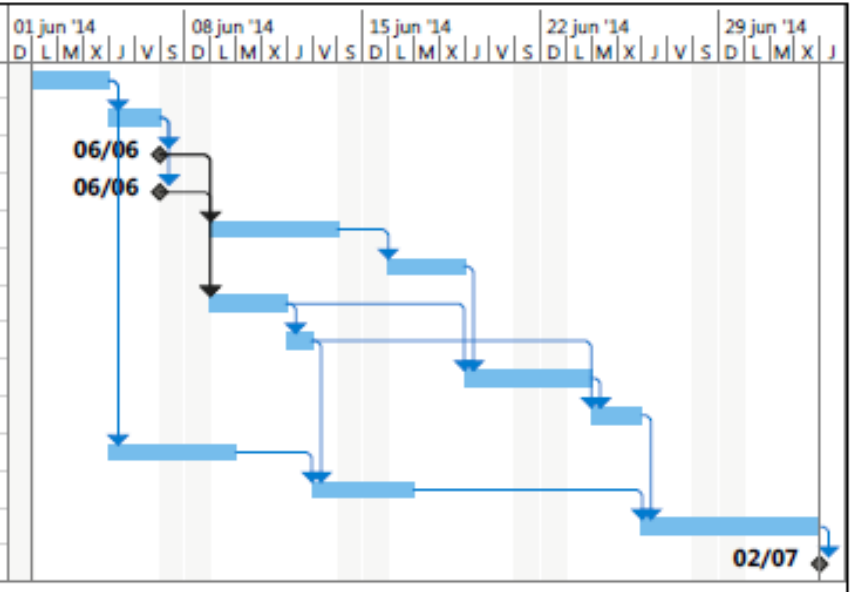


FEATURE	CANTIDAD
PROCESADORES	2 x Intel E5-2665 (Total 16 cores)
RAM	256 GB
DISCO	3 x 1TB 7.2K SAS HDDs
CONEXION A RED	4 X GbE
SOPORTE	3 años 7 x 24
PRECIO DE REFERENCIA	U$S 17K

SAS LASR ANALYTIC SERVER

PLAN TENTATIVO DE IMPLEMENTACION DE LABORATORIO ANALÍTICO SAS SOBRE HADOOP

Id	Nombre de tarea	Comienzo	Duración
1	Definir información a cargar y diseñar tablas en LASR	lun 02/06/14	3 días
2	Hacer sizing formal de plataforma	jue 05/06/14	2 días
3	Disponibilidad de servidor SAS y servidores cluster	vie 06/06/14	0 días
4	Disponibilidad software SAS	vie 06/06/14	0 días
5	Instalar Linux y demás componentes de software de	lun 09/06/14	5 días
6	Instalar distribución seleccionada de Hadoop	lun 16/06/14	3 días
7	Instalar software SAS Server	lun 09/06/14	3 días
8	Instalar software SAS en estaciones clientes	jue 12/06/14	1 día
9	Instalar LASR server en cluster Hadoop	jue 19/06/14	3 días
10	Prueba básica de SAS con Hadoop	mar 24/06/14	2 días
11	Diseñar ETLs	jue 05/06/14	3 días
12	Verificar conectividad a fuentes de datos	vie 13/06/14	2 días
13	Desarrollar carga de datos en VA Data Builder o	jue 26/06/14	5 días
14	Comienzo funcionamiento laboratorio analítico sobre	mié 02/07/14	0 días



LE PROPONEMOS COMENZAR...

Big Data / Hadoop:
Nueva tecnología para resolver grandes problemas de negocio a bajo costo

SAS sobre Hadoop:
SAS adopta esta tecnología en forma integral, minimizando su dificultad de implementación

SAS le propone comenzar su proyecto de Big Data y crecer en forma incremental con SAS sobre Hadoop

¿COMENZAMOS EL PROYECTO ANALYTICS SOBRE HADOOP CON SAS?