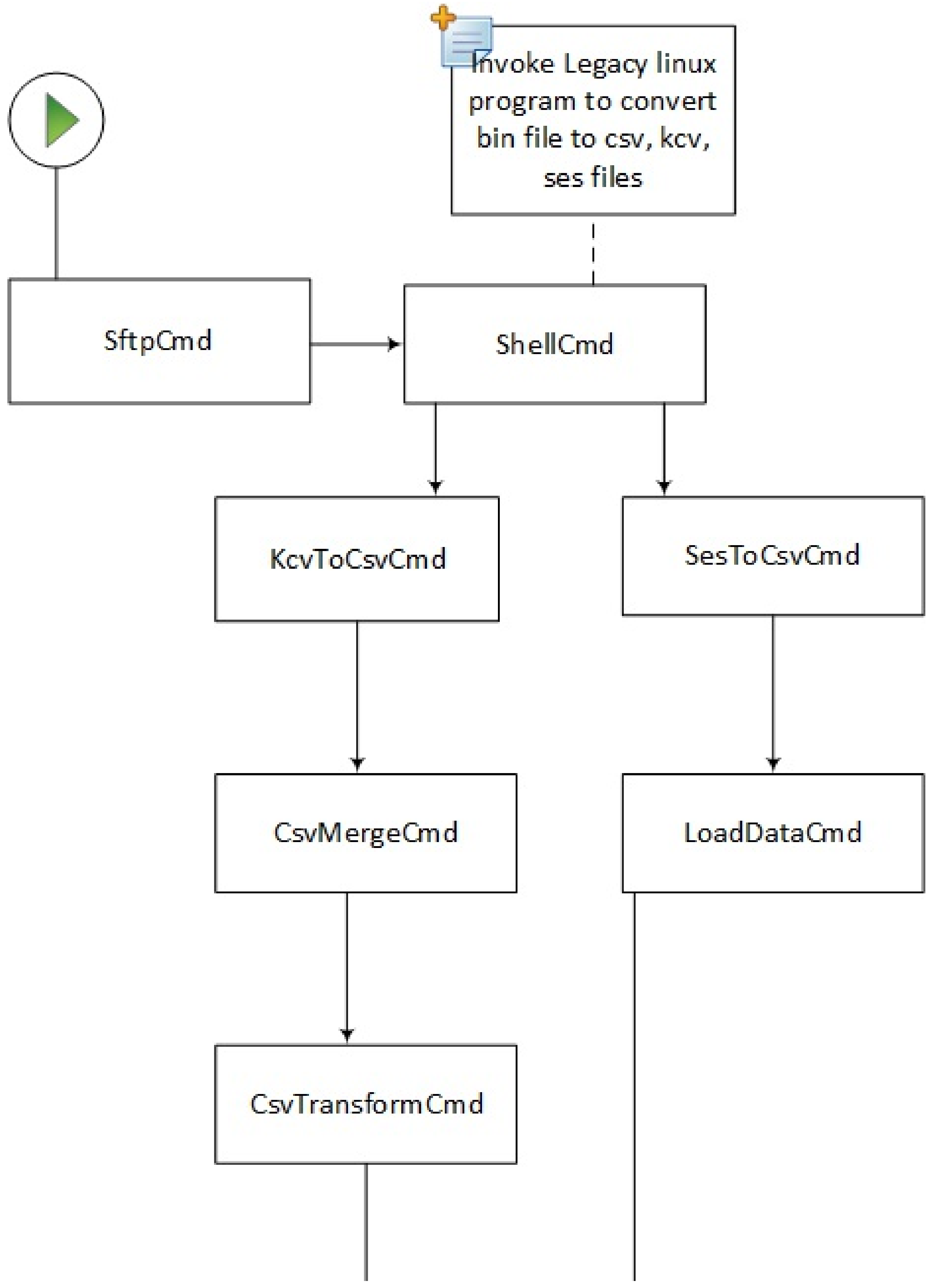
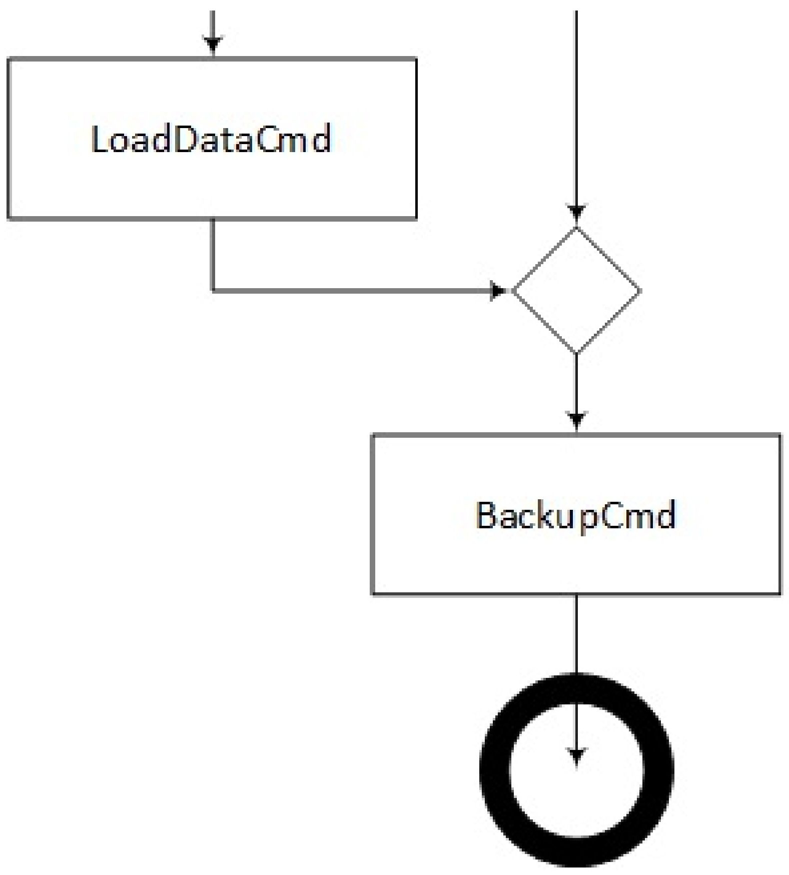
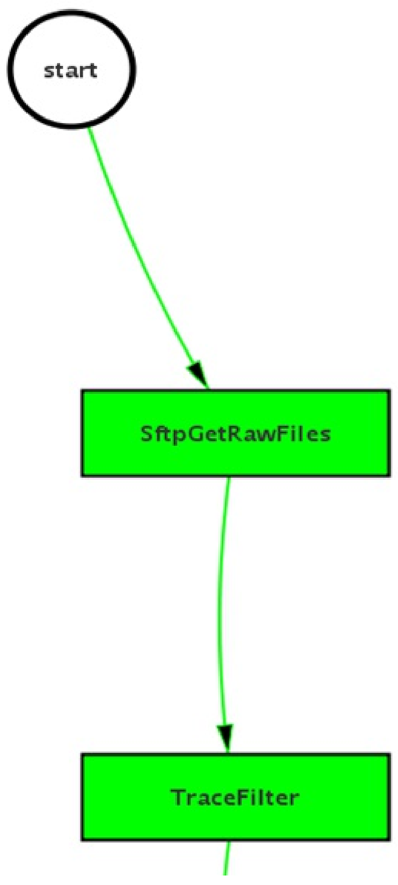
Big Data Analysis Platform

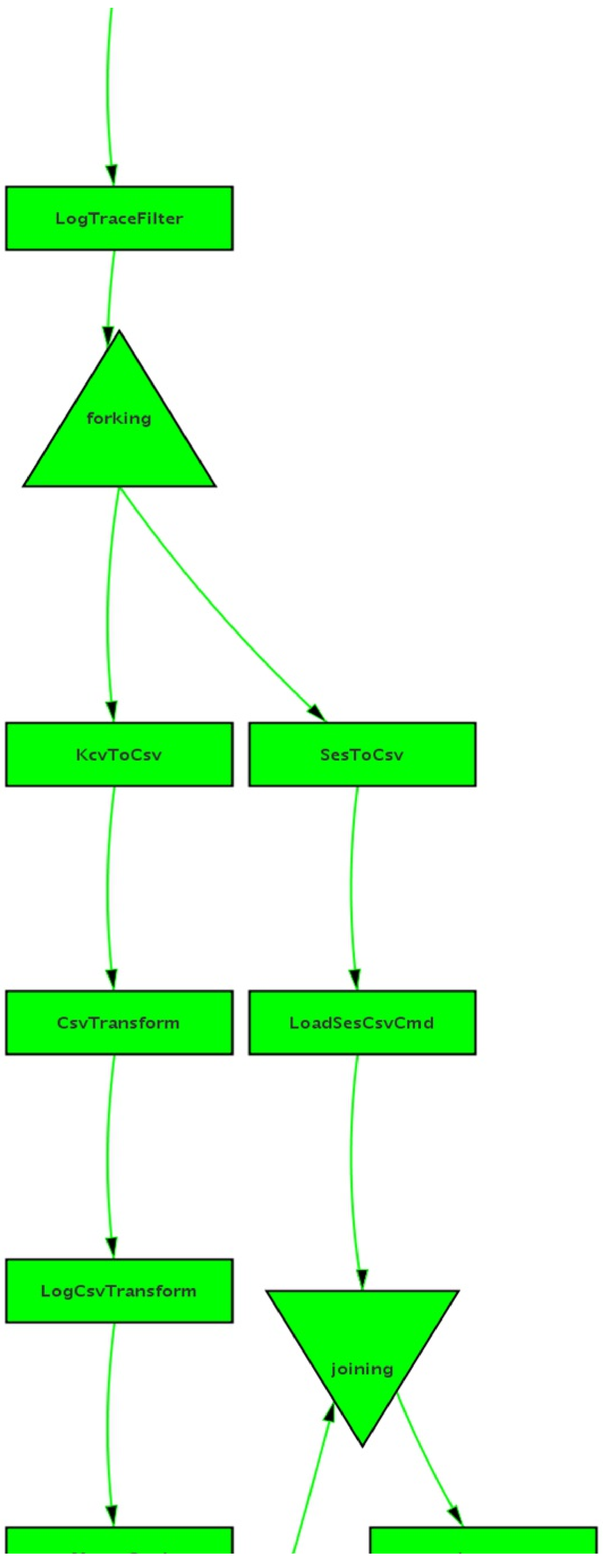
Unified Flow Definition Engine

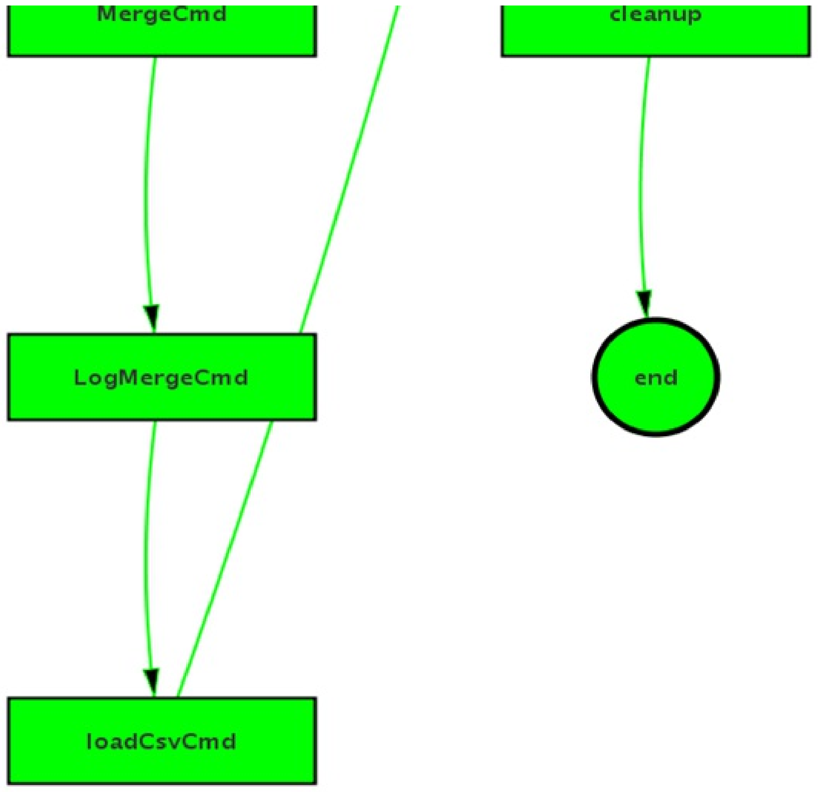




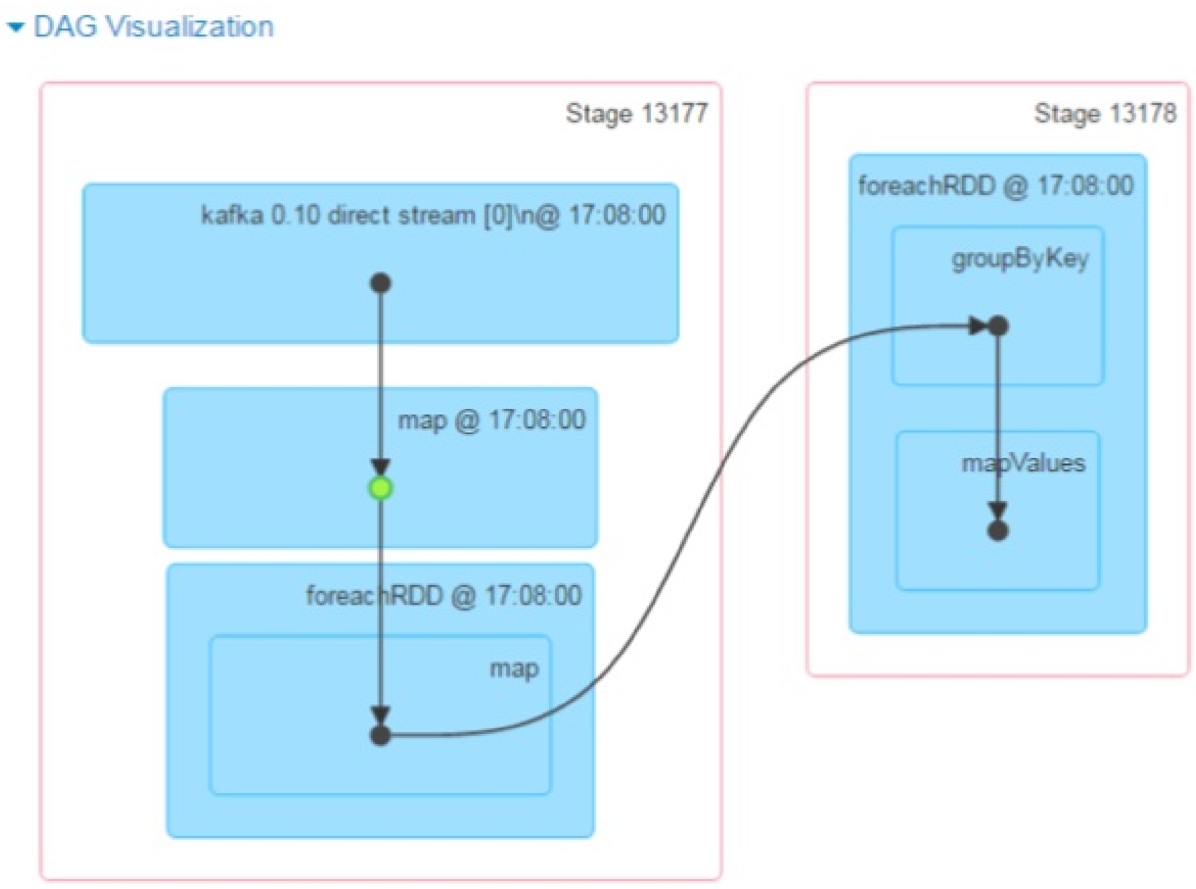
Support Batch Execution on Oozie



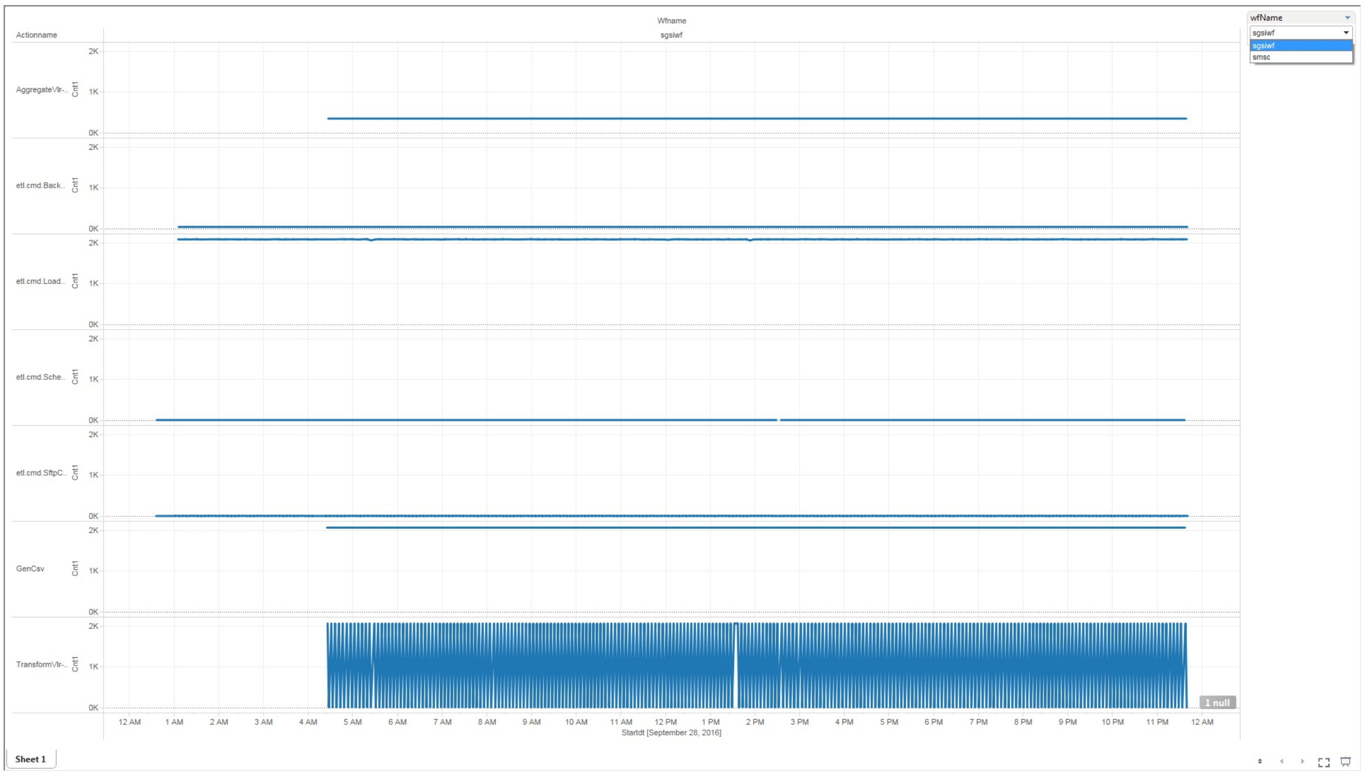

LogTraceFilter
forking
KcvToCsv
SesToCsv
CsvTransform
LoadSesCsvCmd
LogCsvTransform
joining



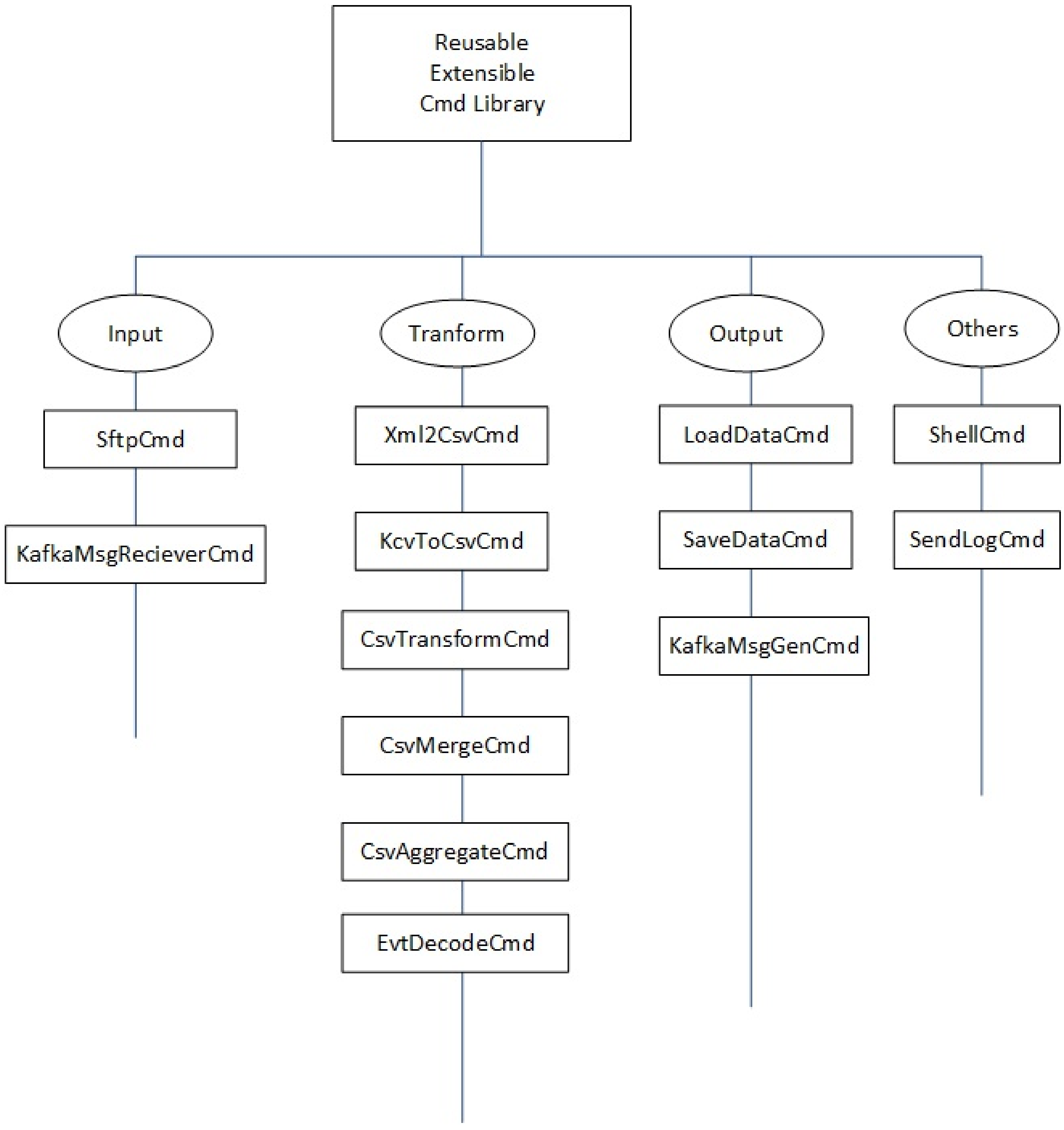
Support Stream Execution on Spark



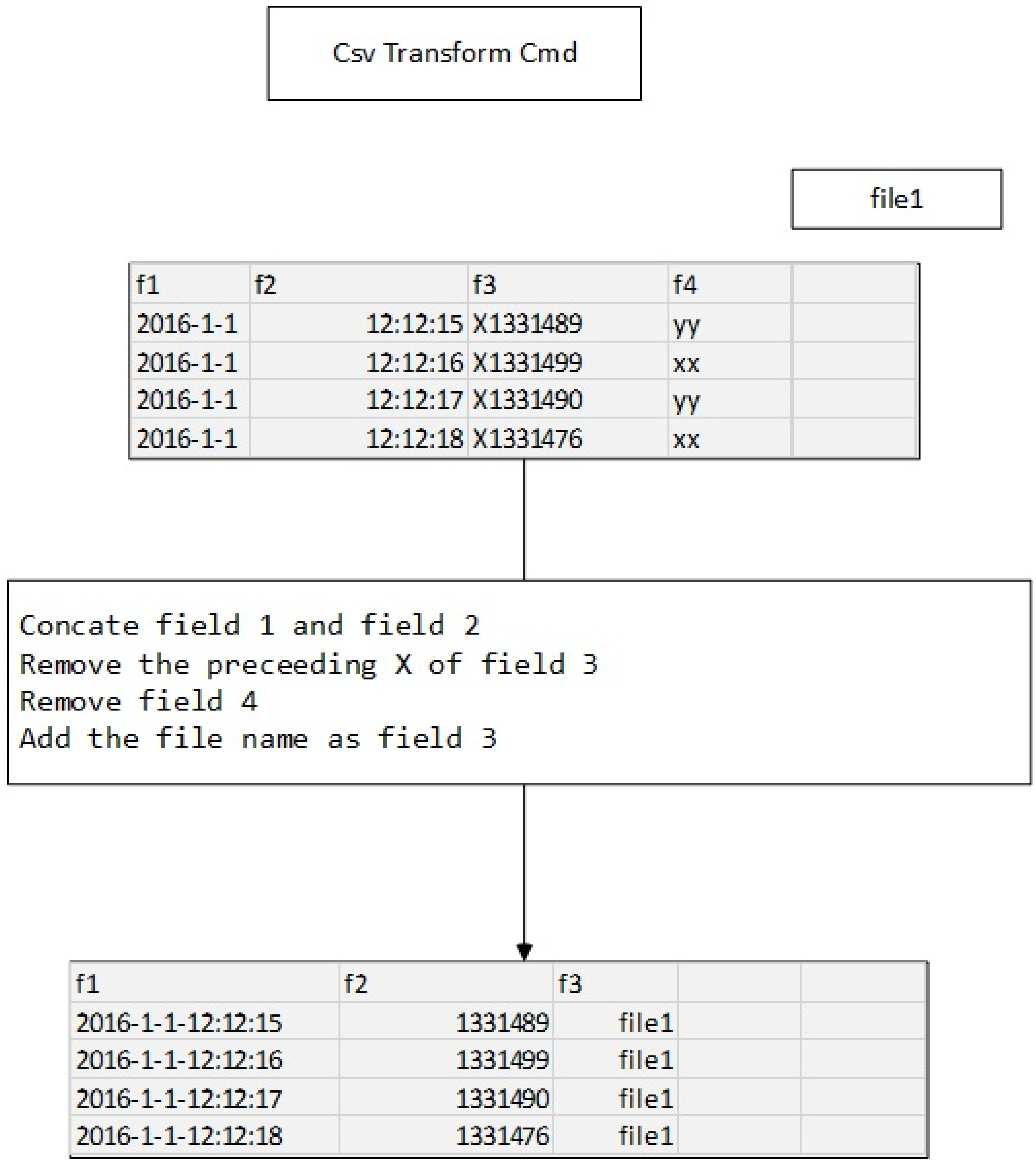
Workflow Monitoring



Reusable, Extensible Cmd Library



1. CSV Transformation Cmd



f1	f2	f3	f4	
2016-1-1	12:12:15	X1331489	yy	
2016-1-1	12:12:16	X1331499	xx	
2016-1-1	12:12:17	X1331490	yy	
2016-1-1	12:12:18	X1331476	xx	

```
Concate field 1 and field 2
Remove the preceeding X of field 3
Remove field 4
Add the file name as field 3
```

f1	f2	f3		
2016-1-1-12:12:15	1331489	file1		
2016-1-1-12:12:16	1331499	file1		
2016-1-1-12:12:17	1331490	file1		
2016-1-1-12:12:18	1331476	file1		

User can specify following column operations (Update, Split, Remove) on the fields for each line of the csv file.

Columns Update

User can merge a range of columns by specifying the merge expression.

```
Example 1:
```

```
col.op=u|68:(fields[68].concat('-')).concat(fields[69])
```

```
update column 68 as join column 68, '-', and column 69

Example 2:
```

```
col.op=u|0:(Number(fields[0])*7*24*3600 + Number(fields[1
])).toString()
```

```
column 0 is number of week since epoch, and column 1 is t
he time part
this update expression caculate the epoch time by merging
 these two columns
```

Column Split

User can split specific column by specify the separator.

```
Example 1:
```

```
col.op=s|3:.
```

```
Split the column 3 into multiple fields by separator '.'
```

Column Remove

```
Example 1:
```

```
col.op=r|3:
```

```
Remove column 3
```

Skip header

Some input files comes with the header line, we need to skip that line for output.

```
Example 1:
```

```
skip.header=true
```

Row Ends with comma

Some input csv files, each line comes with an ending comma, we need to tell Command about this.

```
Example 1:
```

```
input.endwithcomma=true
```

Row Validation

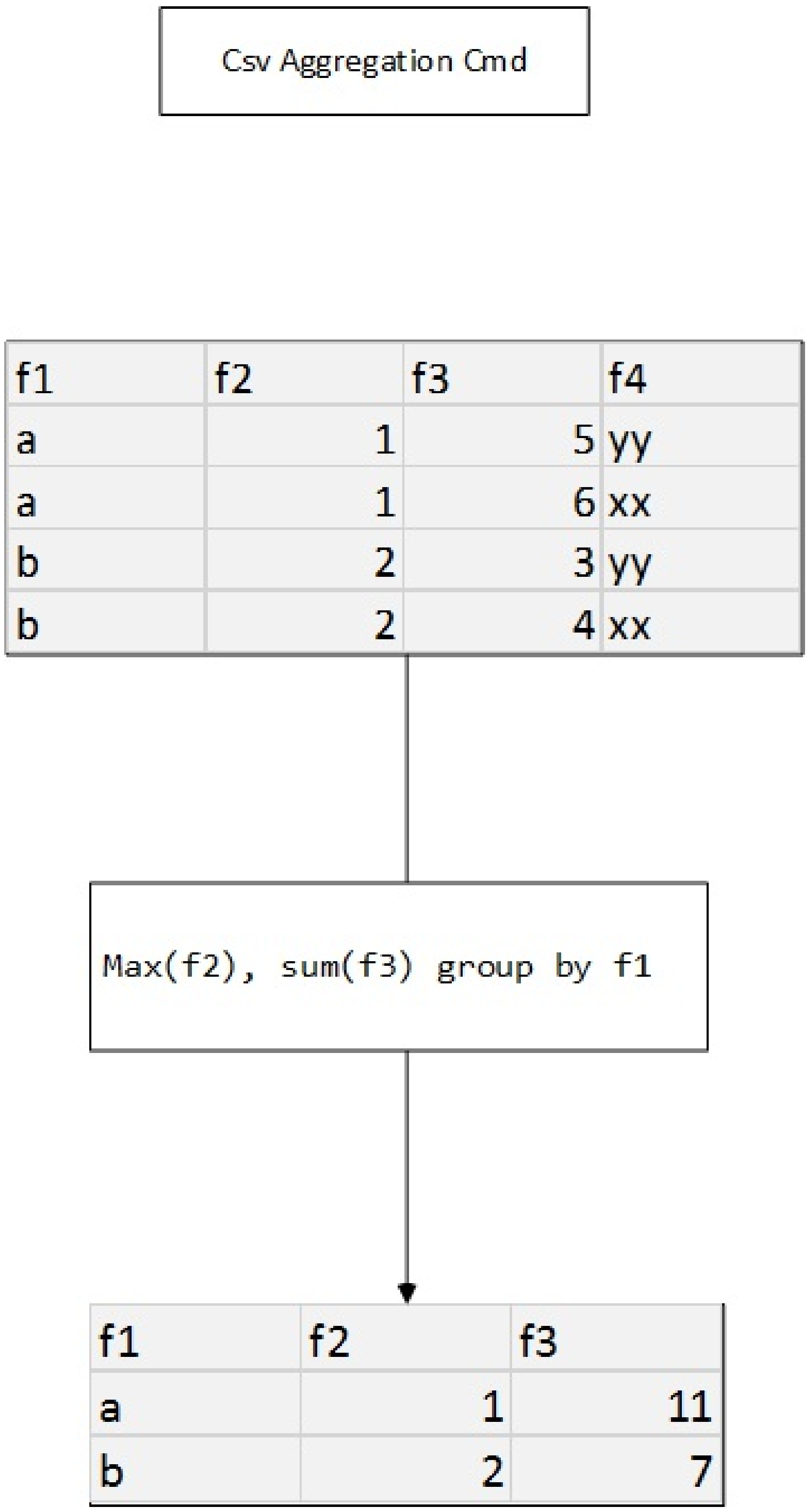
Some input csv files has corrupted lines not intended to be sent to output. We enable user to specify the row validation expression to validate each line

```
Example 1:
```

```
row.validation=fields.length>10
```

```
For this Cmd, the system varaible "fields" (an array of fields for each line) is passed.
```

2. Csv Aggregation Cmd

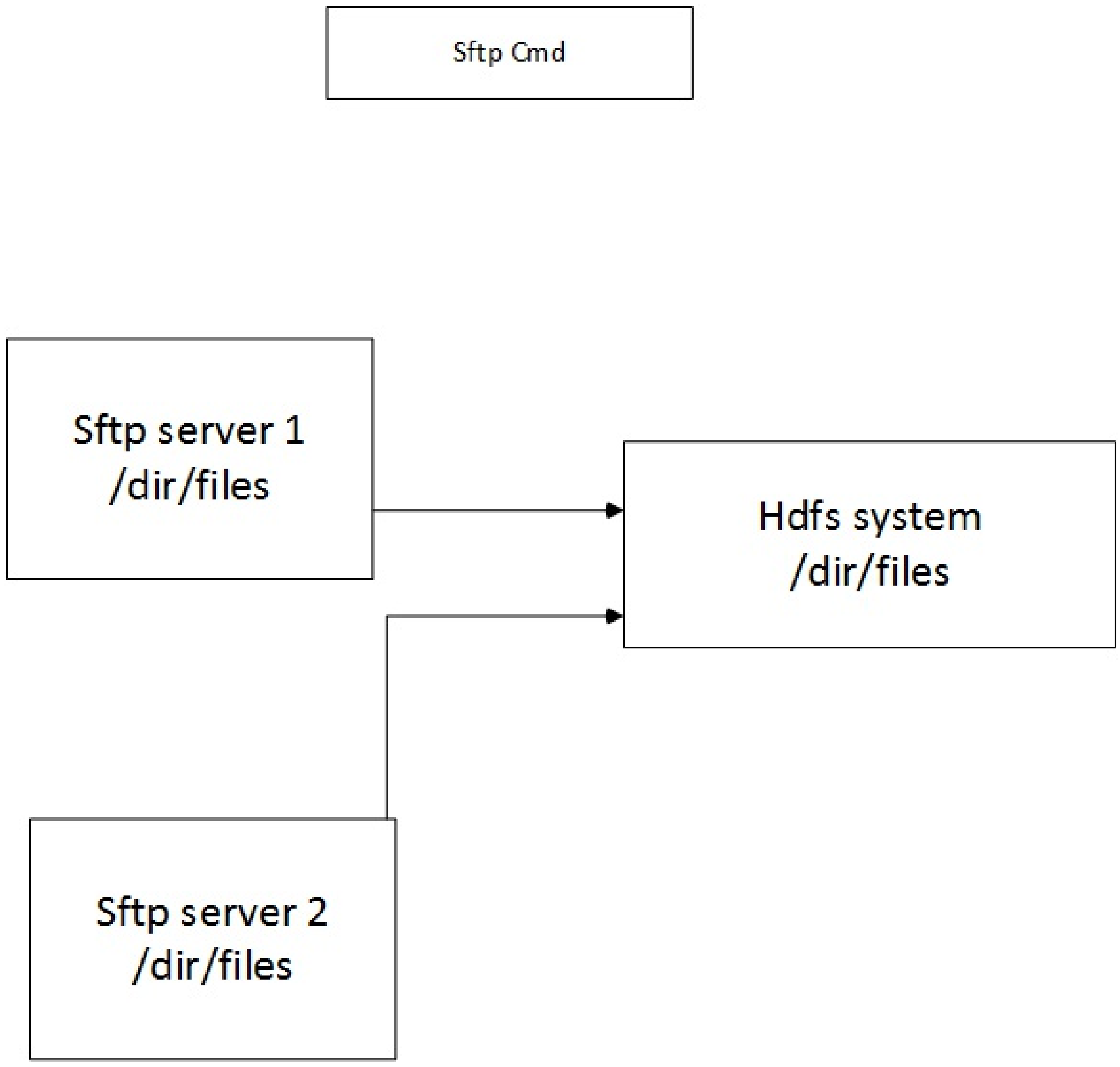


f1	f2	f3	f4
a	1	5	yy
a	1	6	xx
b	2	3	yy
b	2	4	xx

```
Max(f2), sum(f3) group by f1
```

f1	f2	f3
a	1	11
b	2	7

3. SFTP fetch files Cmd



copy files to hdfs from sftp servers

Example 1:

```
incoming.folder='/test/sftp/incoming/'
sftp.user=username
sftp.port=22
sftp.pass=password
sftp.folder=/data/mtccore/sftptest/
sftp.clean=true
sftp.getRetryTimes=3
sftp.connectRetryTimes=3
file.limit=1000
```

4. Schema Update from XMl Cmd

Generate or update the schema based on the input xml files.

Example 1:

Configuration

```
#xml input folder
xml-folder='/test/dynschemacmd/input/'
#csv output folder
csv-folder=/test/dynschemacmd/output/
#schema file
schema.file=/test/dynschemacmd/schema/schemas.txt
#schema history folder, any updated or new schema generat
ed will be put here with timestamp
schema-history-folder=/test/dynschemacmd/schemahistory/
#db schema name
prefix=sgsiwf

FileSystemAttrs.xpath=/measCollecFile/fileHeader/fileSend
er/@localDn
```

```
FileSystemAttrs.name=SubNetwork,ManagedElement
FileSystemAttrs.type=varchar(70),varchar(70)
TableSystemAttrs.xpath = ./granPeriod/@endTime,./granPeri
od/@duration
TableSystemAttrs.name = endTime, duration
TableSystemAttrs.type = TIMESTAMP WITH TIMEZONE not null,
 varchar(10)

xpath.Tables = /measCollecFile/measData/measInfo
xpath.TableRow0 = measValue[1]
TableObjDesc.xpath = ./@measObjLdn
TableObjDesc.skipKeys=Machine,UUID,PoolId,PoolMember
TableObjDesc.useValues=PoolType
xpath.TableAttrNames = ./measType
xpath.TableRows = ./measValue
xpath.TableRowValues = ./r
```

Input Xml:

```xml
<measCollecFile>
    <fileHeader fileFormatVersion="32.401 V5.0" vendorNam
e="Alcatel-Lucent" dnPrefix="">
        <fileSender localDn="SubNetwork=vQDSD0101SGS-L-AL
-20,ManagedElement=lcp-1" elementType="GmscServer,Vlr"/>
        <measCollec beginTime="2016-03-09T07:45:00+00:00"
/>
    </fileHeader>
    <measData>
        <managedElement localDn="SubNetwork=vQDSD0101SGS-L-AL-20,ManagedElement=lcp-1" userLabel="" swVersion="R33.11.00"/>
        <measInfo>
            <granPeriod duration="PT300S" endTime="2016-03-09T07:50:00+00:00"/>
            <measType p="1">VS.avePerCoreCpuUsage</measType>
            <measType p="2">VS.peakPerCoreCpuUsage</measType>
            <measValue measObjLdn="Machine=vQDSD0101SGS-L-AL-20-CDR-01, UUID=a040d711-7ec2-4a5c-be90-6c7f82a3fe21, MyCore=0">
                <r p="1">2.59</r>
                <r p="2">9.13</r>
            </measValue>
            <measValue measObjLdn="Machine=vQDSD0101SGS-L-AL-20-CDR-01, UUID=a040d711-7ec2-4a5c-be90-6c7f82a3fe21, MyCore=1">
                <r p="1">2.26</r>
                <r p="2">8.83</r>
            </measValue>
        </measInfo>
    </measData>
</measCollecFile>
```

Table Name

The table name is defined by “TableObjDesc”. the evaluated value of the TableObjDesc.xpath is a csv string, composed of different attributes, in this example, it is evaluated to “Machine=vQDSD0101SGS-L-AL-20-CDR-01, UUID=a040d711-7ec2-4a5c-be90-6c7f82a3fe21, MyCore=0”

TableObjDesc.skipKeys defined which keys are omitted in the table name composition, in this example “Machine”, “UUID” are skipped, so the table name is “MyCore_”.

Table Fields

The fields of each table is composed of following 4 groups of fields:

1. System Attributes from File Scope: defined by “FileSystemAttrs”

In this example the xpath is defined as “/measCollecFile/fileHeader/fileSender/@localDn”, it is evaluated to “SubNetwork,ManagedElement”

2. System Attributes from Table Scope: defined by “TableSystemAttrs”

In this example the xpath is defined as “./granPeriod/@endTime,./granPeriod/@duration”, they are “duration”,“endTime”

3. Table Object Description Fields: defined by “TableObjDesc”

In this example the xpath is defined as “./@measObjLdn” within xpath.Tables (which is defined as “/measCollecFile/measData/measInfo” in this example.), they are evaluated to “Machine=vQDSD0101SGS-L-AL-20-CDR-01, UUID=a040d711-7ec2-4a5c-be90-6c7f82a3fe21, MyCore=0”, and the fields will be added are “Machine,UUID, MyCore”.

4. Table-wise attributes: defined by “xpath.TableAttrNames”

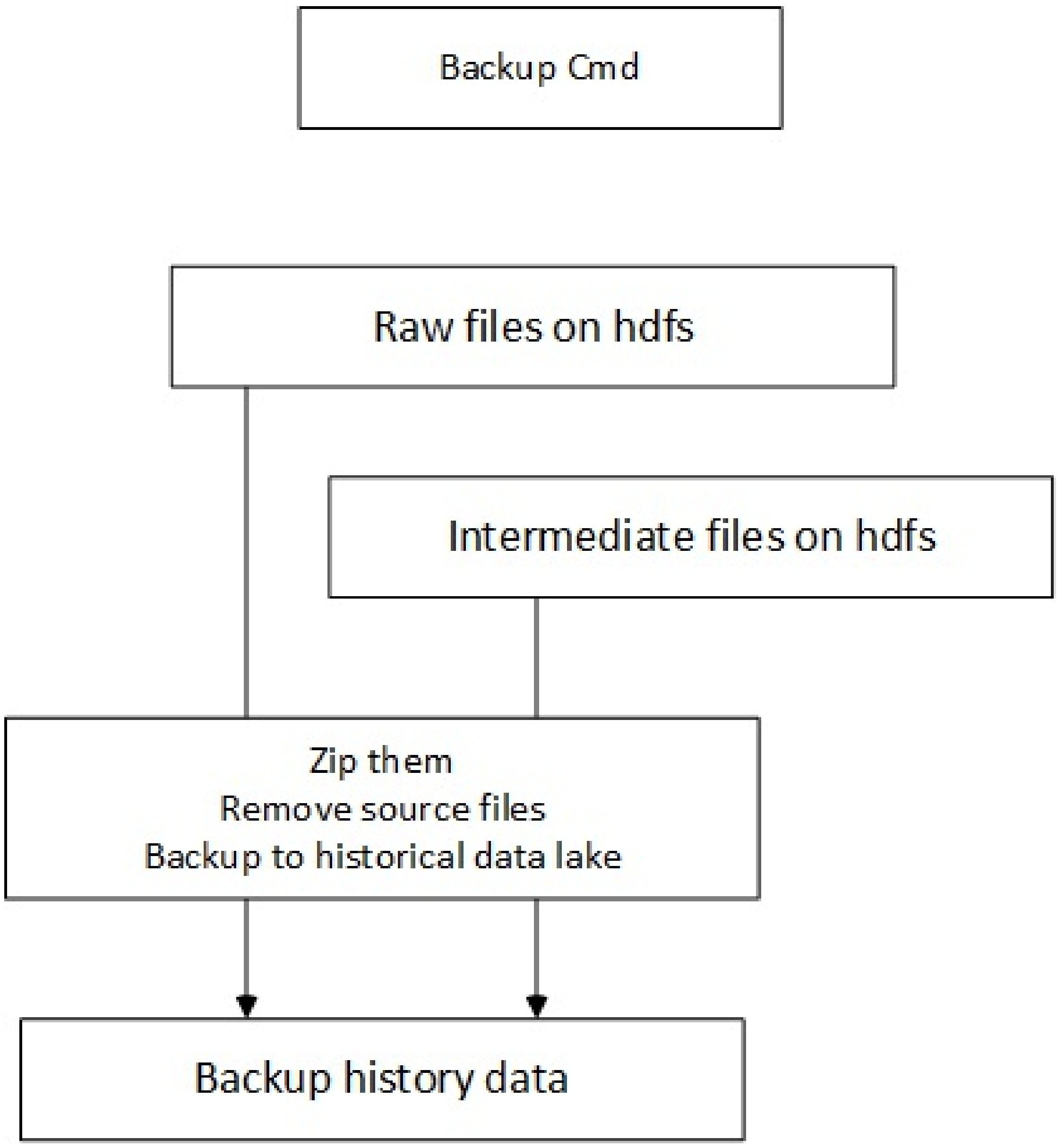
In this example, defined by “./measType” within xpath.Tables (which is defined as “/measCollecFile/measData/measInfo” in this example.), they are evaluated to “VS.avePerCoreCpuUsage,VS.peakPerCoreCpuUsage”.

So following table will be created if not already exist in the schema:

```
create table sgsiwf.MyCore_(
endTime TIMESTAMP WITH TIMEZONE not null,
duration varchar(10),
SubNetwork varchar(70),
ManagedElement varchar(70),
Machine varchar(54),
MyCore numeric(15,5),
UUID varchar(72),
VS_avePerCoreCpuUsage numeric(15,5),
VS_peakPerCoreCpuUsage numeric(15,5));
```

5. Xml To Csv Cmd

6. Backup Cmd



zip all the raw files and intermediate files and backup to data lake

Exmaple 1:

```
file.folder='/test/BackupCmd/data/allFolder1/','/test/Bac
kupCmd/data/wfidFolder1/'
file.filter='.*',WFID+'.*'
data-history-folder=/test/datahistory/
```

user can specify a list of folder filters and file filters, 1 to 1 mapped.

This cmd will zip all these files specified in a zip file named as wfid.zip under the data-history folder. Then it will remove them.

7. KCV to CSV Transformation Cmd

The key colon value format to csv format transformation. static configuration:

```
record.start=^TIME:.* UNCERTAINTY:.*
#value,key regexp for the kcv format
record.vkexp=[\\s]+([A-Za-z0-9\\,\\. ]+)[\\s]+([A-Z][A-Z
]+)
record.fieldnum=8
```

sample input:

```
TIME: 1815,449649.119 UNCERTAINTY: 0.000 SOURCE: external
 Primary ID: X310007204992127F TYPE: standard fix SESSION
```

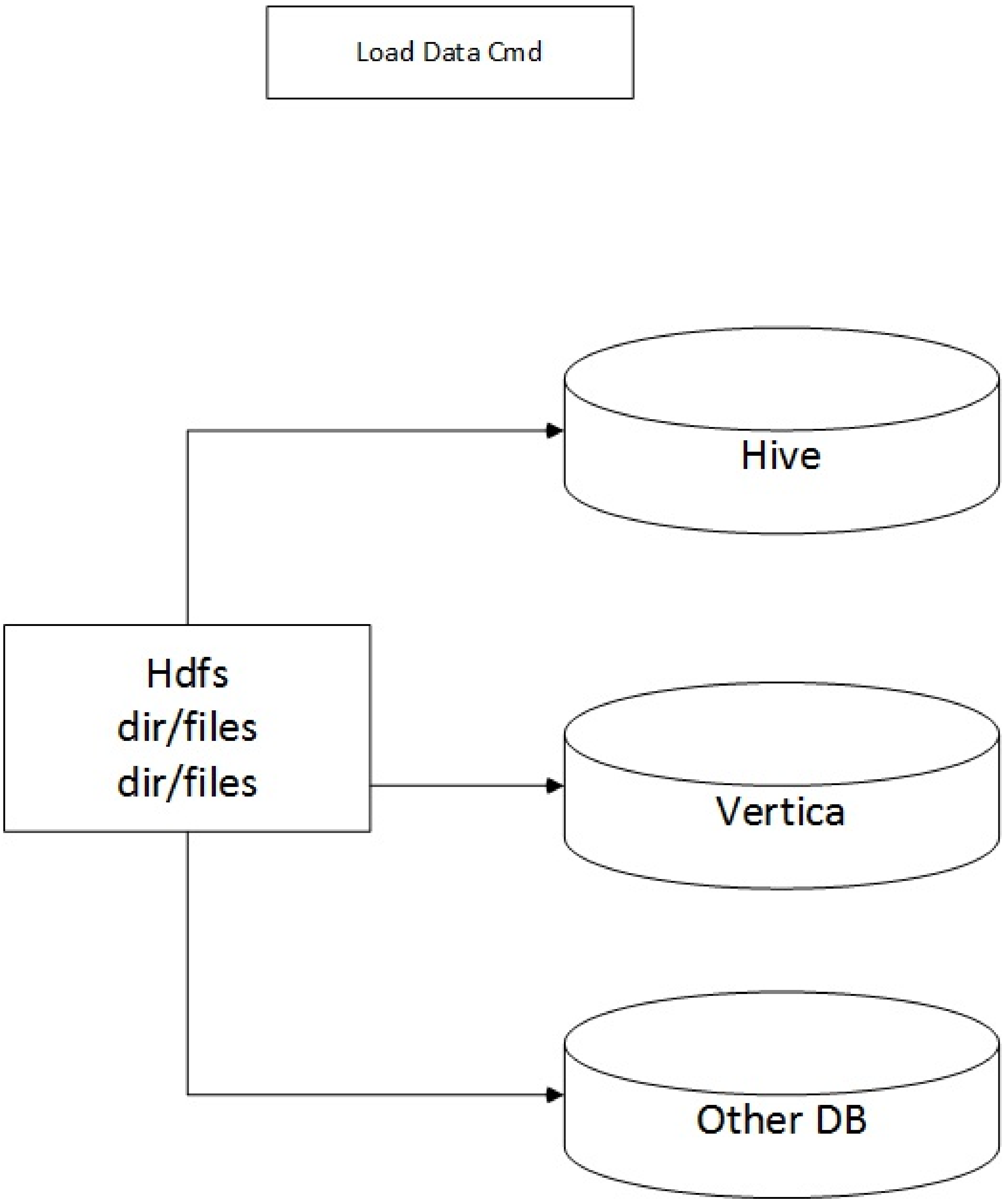
```
: 202967223 APPLICATION: 327897
POSITION ENGINE: integrated RESULT: success   GPS: 0 AFLT
: 7 EFLT: 0 ALTITUDE: 1 ORTHO: 0 TIME AID: 0 RTD: 0 STB:
0 POS: 0
```

sample output:

```
1815,449649.119,0.000,external Primary,X310007204992127F,
standard fix,202967223,327897,,
1815,449648.824,0.000,external Primary,X310005414400271F,
standard fix,202967224,262221,,
```

1. record.start specify the regular expression to identify the begining of a record.
2. record.vkexp specify the value key regular expression to match the value and key
3. record.fieldnum specify the number of fields to extract for each record

8. Load Database Cmd



load the csv files from dfs to database.

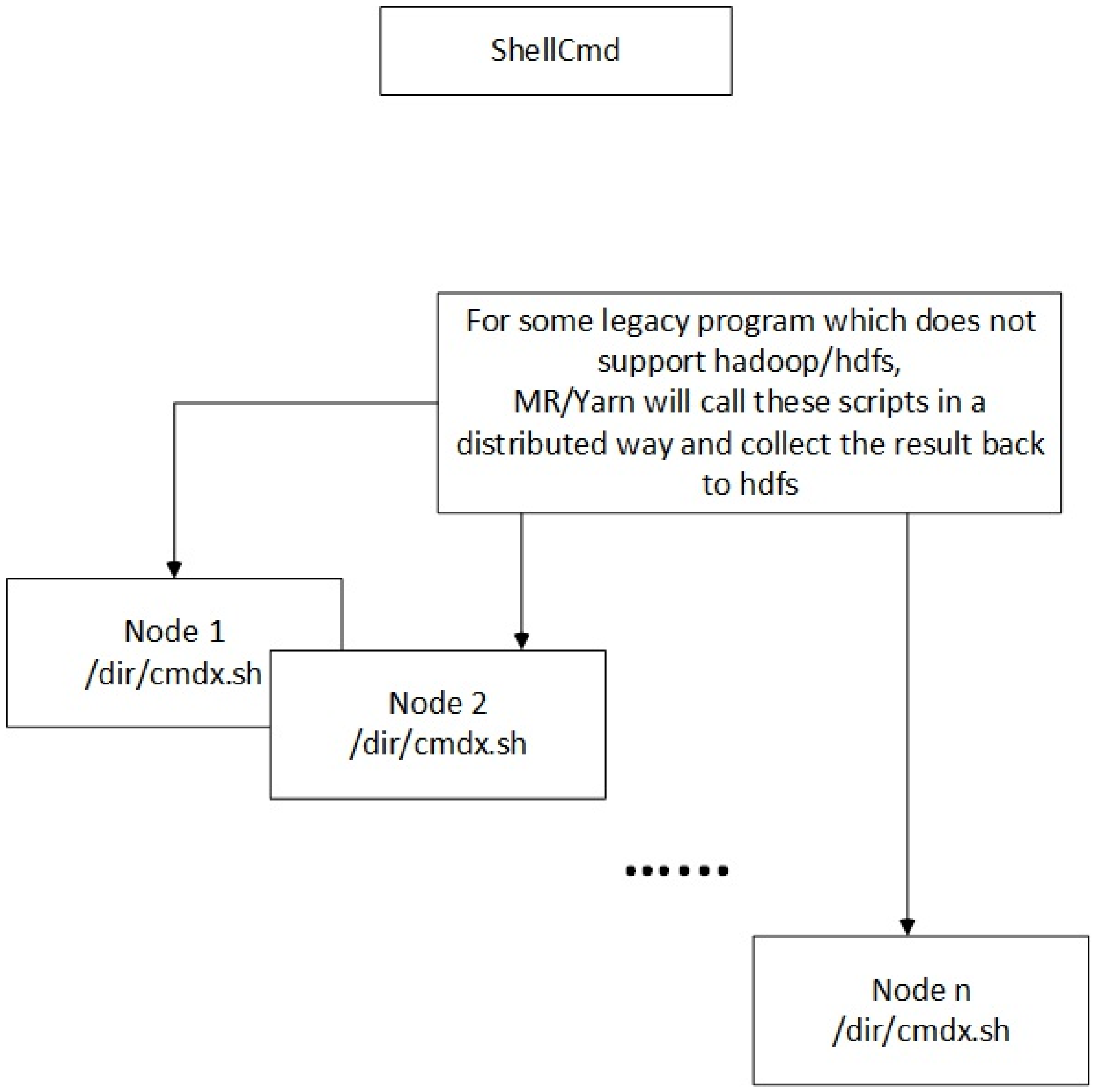
This cmd is used to load csv data to a predefined database.

Vertica/Hive are supported.

Example 1: Static Configuration

```
hdfs.webhdfs.root=http://xx.xx.xx.xx:50070/webhdfs/v1
csv.folder=/pde/fixcsv1/
csv.file='/test/loadcsv/input/' + tableName + '.csv'
schema.file=/test/loadcsv/schema/test1_schemas.txt
db.prefix=sgsiwf
db.type=vertica
db.driver=com.vertica.jdbc.Driver
db.url=jdbc:vertica://xx.xx.xx.xx:5433/dbname
db.user=username
db.password=password
db.loginTimeout=35
```

9. Shell Cmd


ShellCmd
For some legacy program which does not support hadoop/hdfs, MR/Yarn will call these scripts in a distributed way and collect the result back to hdfs
Node 1 /dir/cmdx.sh
Node 2 /dir/cmdx.sh
....
Node n /dir/cmdx.sh

Example 1:

```
srcfolder:/tmp/original
destfolder:/tmp/backupfolder
command=bash /tmp/copyfile.sh $srcfolder $destfolder $key
```

invoke the shell commnad via MapReduce.

This cmd will replace the attributes and execute the cmd.

10. Event Decode Cmd

Example 1:

```
#record type specification
#where is the event type
event.idx=5
#event type values
event.types=default,005730,005706

#main message specification
#where is the main message
message.idx=6
#how to extract the fields in the main message
message.fields=IMSI,E164,GTAddr,ReturnCause
#regexp to identify each message field
default.regexp=.+ E.164 ([0-9]+) .+
default.attr=E164
005730.regexp=.+ GT Addr ([0-9]+)
005730.attr=GTAddr
005706.regexp=.+ TimeOut : ([0-9]+) .+ GT Addr ([0-9]+)
005706.attr=IMSI,GTAddr
```

11. Send Log Cmd

Then use the oozie workflow engine and coordination engine to schedule the jobs.

Command Developer Guide

Construct a Cmd

Each Cmd will be initialized by following parameters

wfName

wfid

the wfid stands for workflow instance id. It will be generated to identify a ETL batch process instance, usually that stands for a list of input files (dataset).

config

defaultFs

other arguments

Processing Modes

Single Process Mode

Single JVM process mode.

```
List<String> sgProcess()
```

return list of log info.

Map Mode

Then the cmd needs to implement

```
public Map<String, Object> mapProcess(long offset, String
 row, Mapper<LongWritable, Text, Text, Text>.Context cont
ext) throws Exception;
```

Map Reduce Mode

needs to implement 1 more method, in addition to the one in Map Mode

```
/**
* @return list of newKey, newValue, baseOutputPath
**/
public List<String[]> reduceProcess(Text key, Iterable<Te
xt> values)
```

Spark Reciever Mode

```
public JavaRDD<Tuple2<String, String>> sparkProcess(JavaR
DD<String> input);
```

Spark Process Key Value Mode

```java
public JavaRDD<Tuple2<String, String>> sparkProcessKeyVal
ue(JavaRDD<Tuple2<String, String>> input);
```