Applying Testing Techniques to Hadoo Development

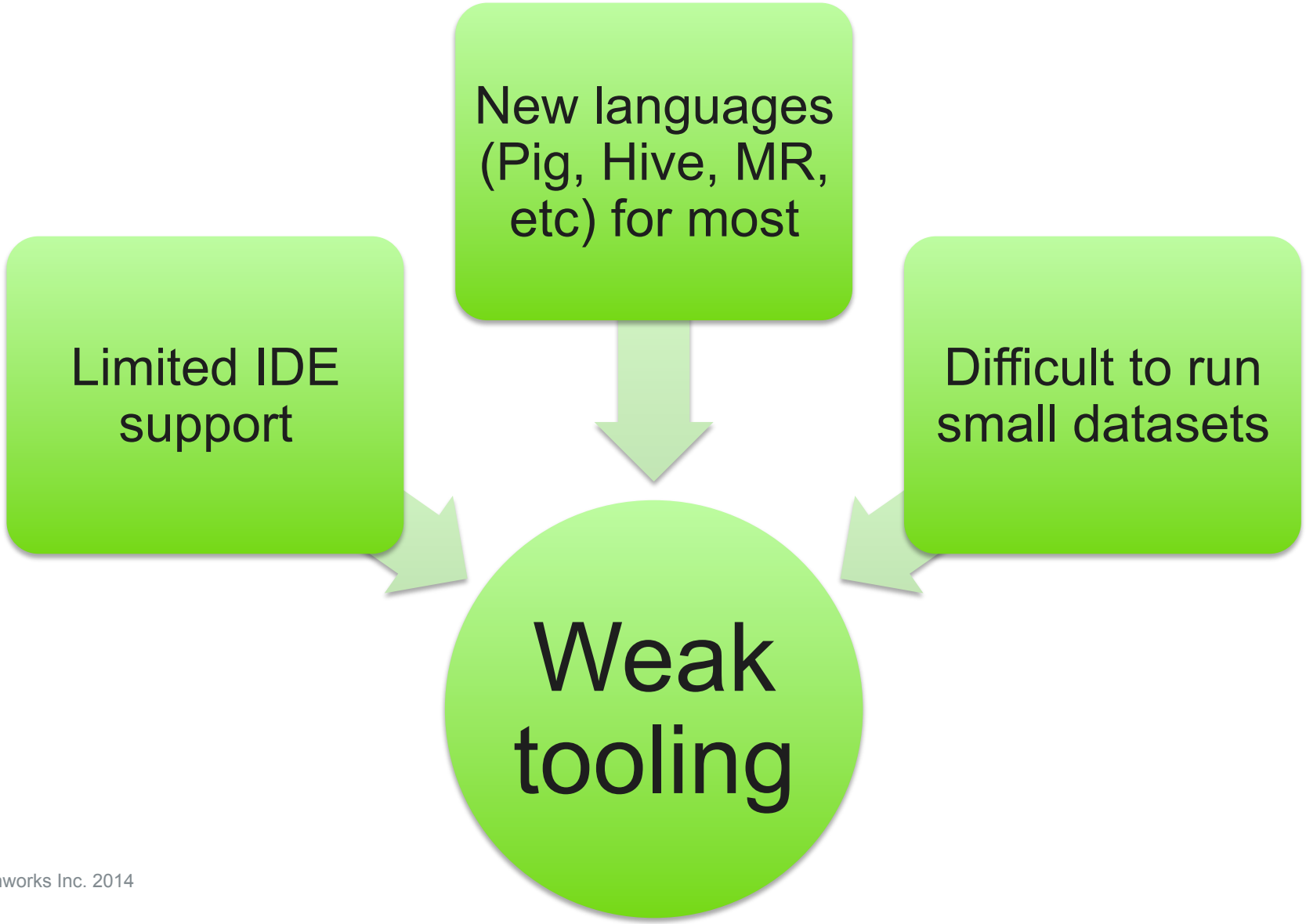
Hortonworks. We do Hadoop.

Presented by Mark Johnson

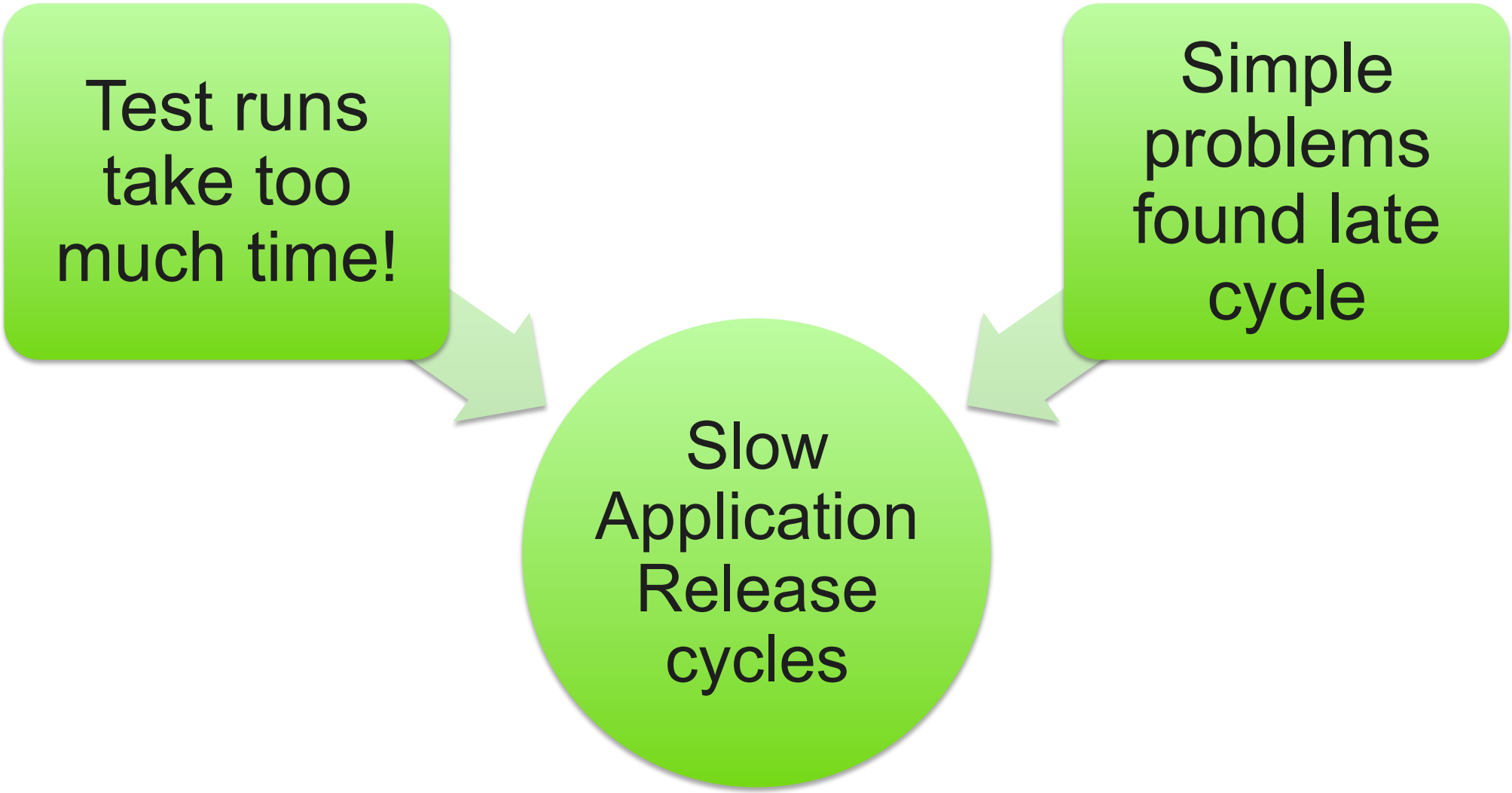
Regional Director Consulting Services

mjohnson@hortonworks.com

Good Hadoop Development is Hard!



Hortonwo



Fast Release Pressure!

BUGS

Will anyone really notice that ittle problem which affects only 5% of the rows?

Yes!

5% of the financial transactions on Wall Street is a lot of money!

Big Data can amplify small problems

How are you currently preventing BUGS?

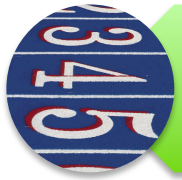
adoop Testing Barriers

I don’t have time to write tests

Sample Dataset is too big or not all test data is in Sample

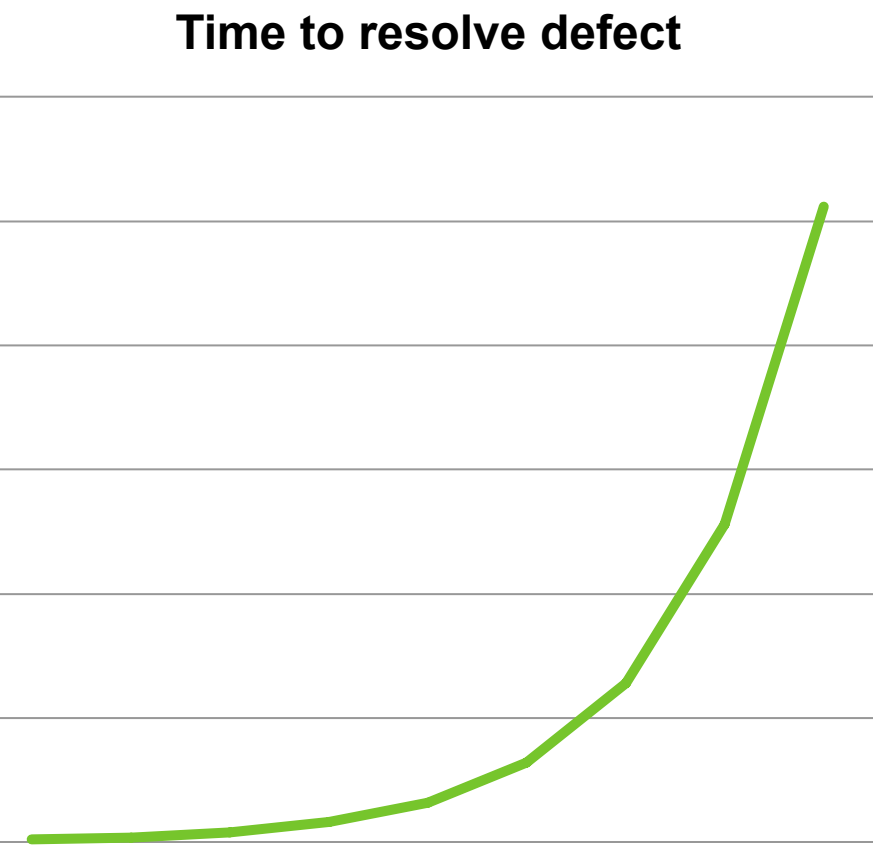
I am not allowed to perform test jobs on the cluster

Tight coupling



Unfamiliar with available testing tools

onsequences



- **1-10-100 rule**
- **Spend more time fixing code than writing new code!**
- **Defects in production mean late night calls and loss of respect!**

Automated Tests define “DONE”

What do we need to test?

Everything we touch

What do we need to check for in our tests?

1. Does it run?

2. Is the code Functionally Correct

- Data filters are accurate
- Loops and branches are accurate
- Calculations are correct

3. Resources are correctly referenced

- Missing files
- Bad file names
- Etc.

4. Exceptions and Errors properly handled

- System left in a safe state
- User / SysAdmin informed of the problem
- Idempotency

ata Permutation Problem

1 table 20 columns and 10 billion rows and

The columns used for filtering each have 100 values

All combinations of the 20 columns and their available values can be used as filter logic for the production queries!

Does this sound familiar?

Could you even create enough tests to validate all the permutations?

esting Rules:

Materiality Rule: Start with high value tests which are easy to implement

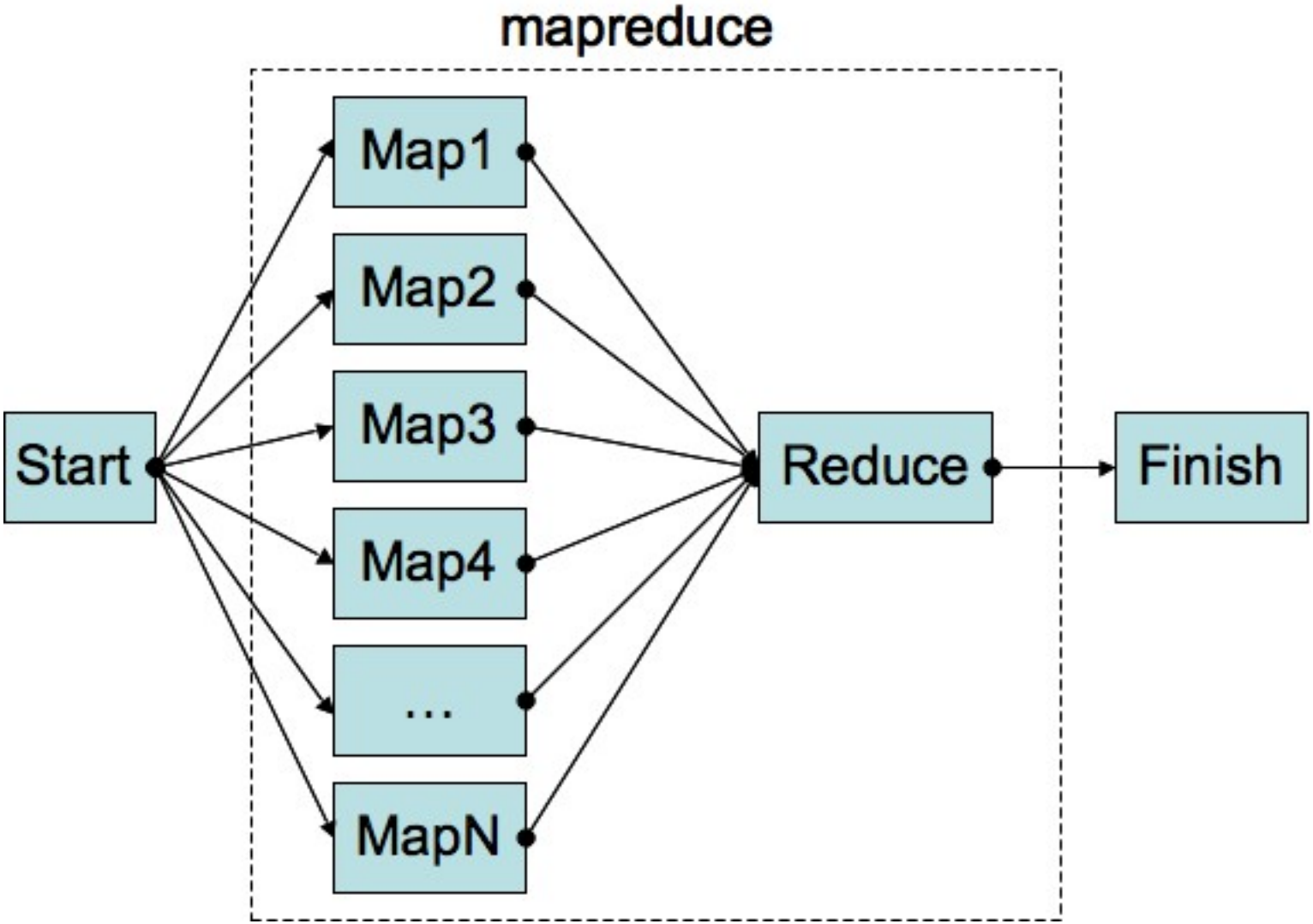
Light & Fast: Don’t create an overly heavy test process

Positive Tests:

- Tests should contain one general data condition
- Individual tests only for those determined by specific data conditions

Negative Tests:

- Data not present



ample MapReduce

```
ic static class MapClass extends MapReduceBase
plements Mapper<LongWritable, Text, Text, IntWritable> {

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

olic void map(LongWritable key, Text value,
              OutputCollector<Text, IntWritable> output,
              Reporter reporter) throws IOException {
String line = value.toString();
StringTokenizer itr = new StringTokenizer(line);
while (itr.hasMoreTokens()) {
  word.set(itr.nextToken());
  output.collect(word, one);
}



 reducer class that just emits the sum of the input values.

ic static class Reduce extends MapReduceBase
plements Reducer<Text, IntWritable, Text, IntWritable> {

olic void reduce(Text key, Iterator<IntWritable> values,
                 OutputCollector<Text, IntWritable> output,
                 Reporter reporter) throws IOException {
int sum = 0;
while (values.hasNext()) {
  sum += values.next().get();
}
output.collect(key, new IntWritable(sum));
```

- Just Java Code
- Can use JUnit with Mocking

Hortonwo

esting: MapReduce

ols:

ings tested:

Allow sample data

Validate map and reduce outputs

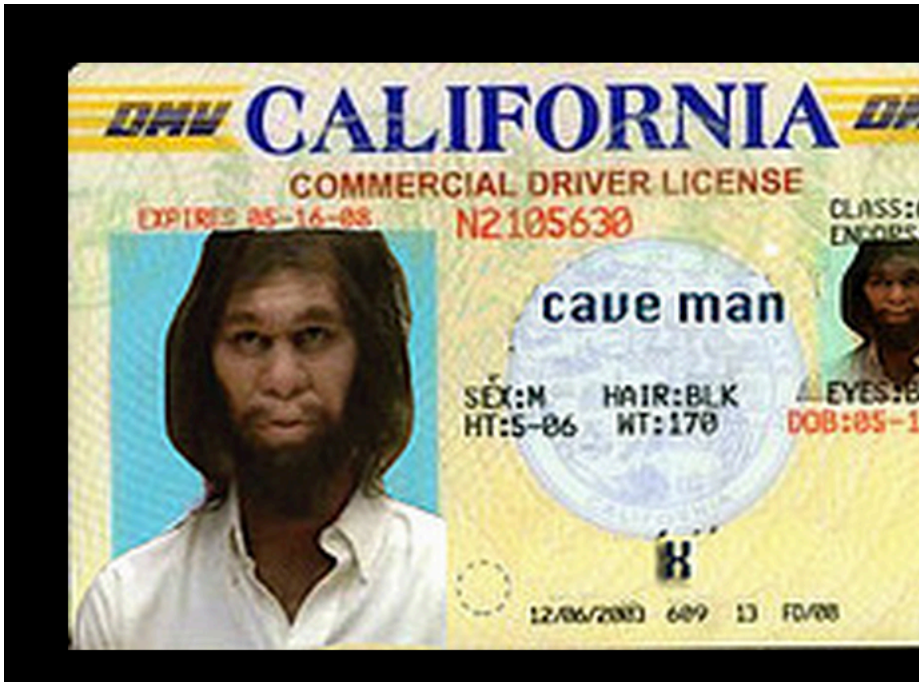
raditional Pig testing tools

DUMP

System.out.println

ILLUSTRATE

DESCRIBE

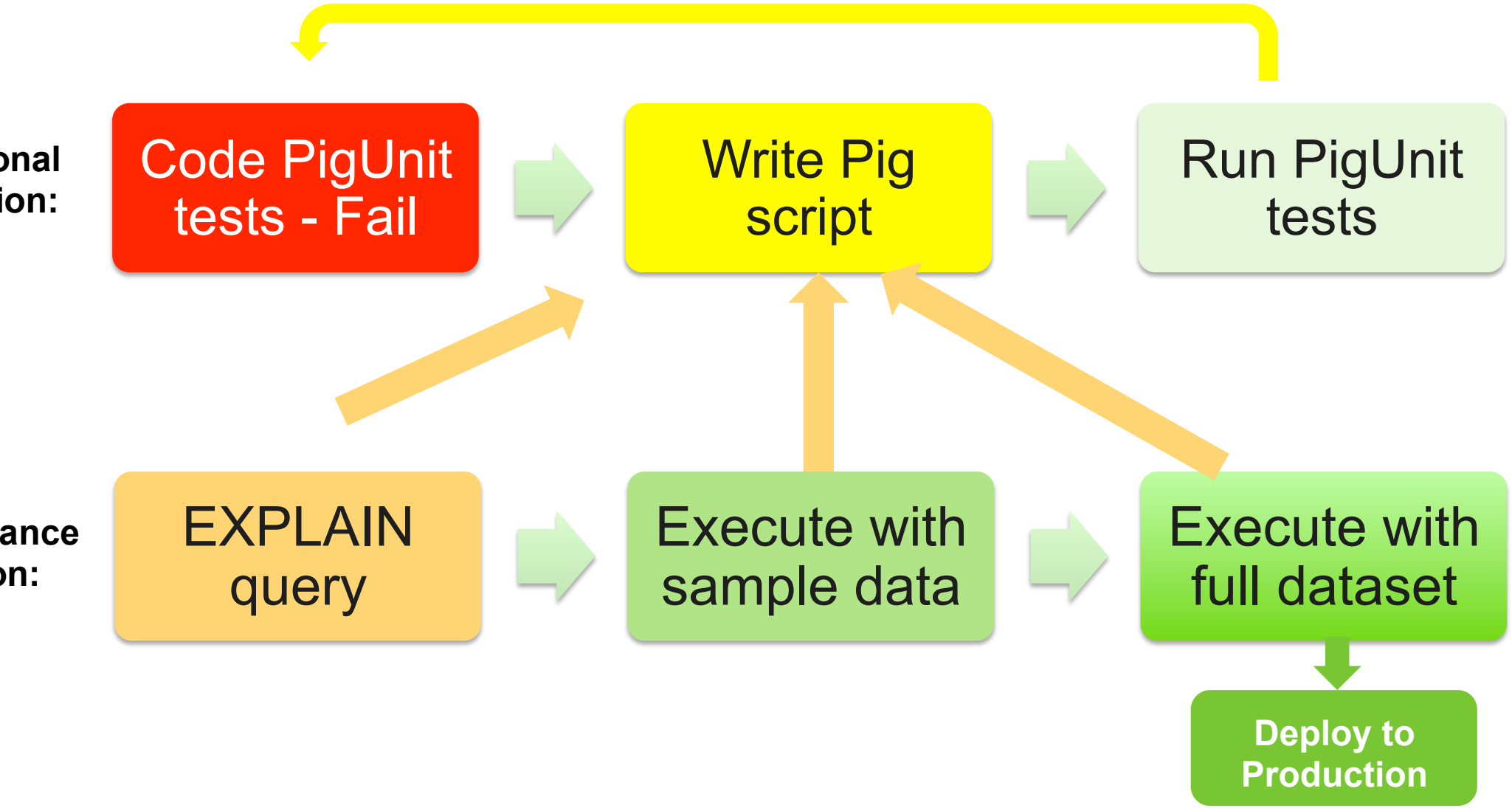


est Results before PigUnit

```
ime in seconds):
    Reduces MaxMapTime      MinMapTIme      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   M
Las    Feature Outputs
15012_0006  1       1       4       4       4       4       4       data,queries_count,queries_group GROUP_BY,COMBINER
15012_0007  1       1       3       3       3       3       4       queries_ordered  SAMPLER
15012_0008  1       1       3       3       3       3       4       queries_ordered  ORDER_BY,COMBINER
15012_0009  1       1       4       4       4       4       4       queries_ordered          hdfs://sandbox.hortonworks.com:80
```

99 Seconds

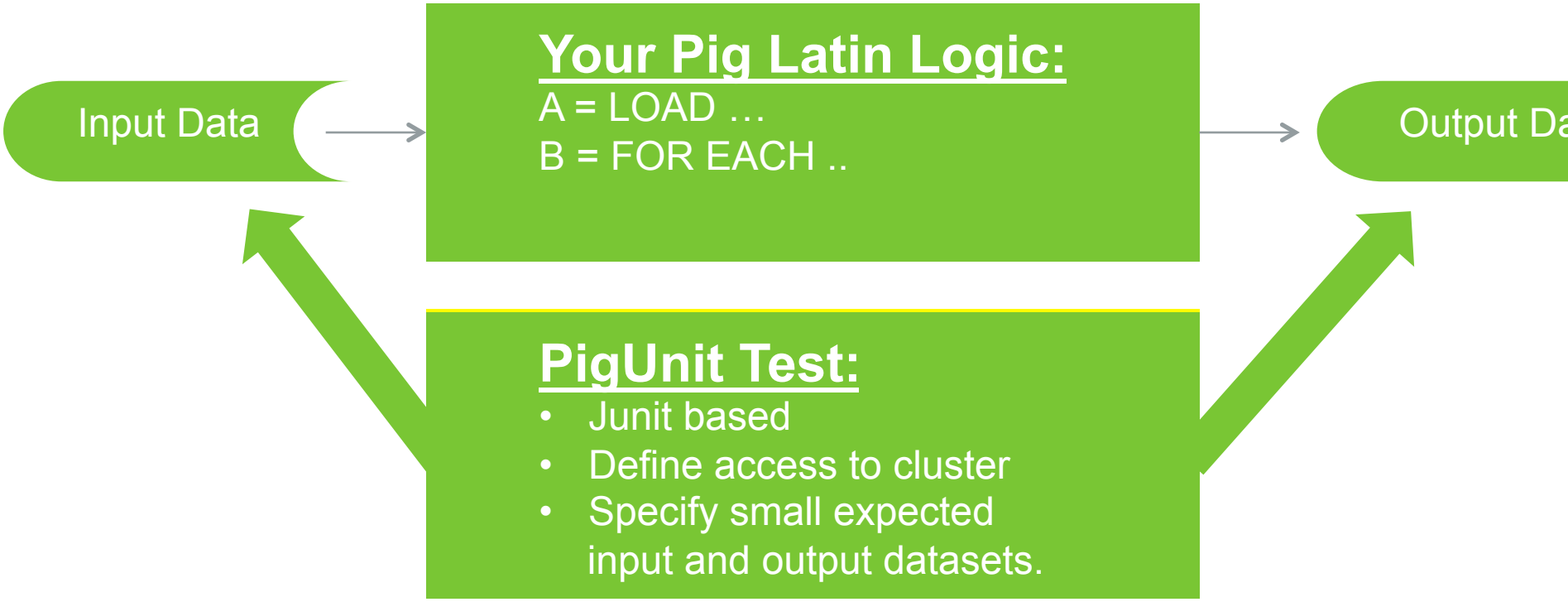
est Driven PIG Development Process



igUnit Script Approach

.pig

ETL.java



Hortonwo

igUnit Example: Requirements

Data to be passed with each service name invocation as a separate lir
Count all web services invoked
Return back only the top two services in descending order

ected Input:

```
ring[] input = {
        "yahoo",
        "yahoo",
        "yahoo",
        "twitter",
        "facebook",
        "facebook",
        "linkedin",
```

Expected Output:

```
String[] output = {
                "(yahoo,3)",
                "(facebook,2
};
```

Hortonwo

igUnit Test

```
c void testTop2Queries() throws Exception {
tring[] args = {
        "n=2",
;

igTest test = new PigTest("top_queries.pig", args);

tring[] input = {
        "yahoo",
```

```
Running example.SimplePigTest
Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.053 sec
Test example.SimplePigTest FAILED
```

```
;

tring[] output = {
        "(yahoo,3)",
        "(facebook,2)",
;

est.assertOutput("data", input, "queries_limit", output);
```


Hortonwo

igUnit Example: The Pig Script

```
a =
 LOAD 'input'
 AS (query:CHARARRAY);

ries_group =
 GROUP data
 BY query;

ries_count = FOREACH queries_group
 GENER  group AS
      ery, SUM(data) AS otal;
```

- Most programming bugs are simple **typos** or **syntax** errors.
- The PigUnit test found this in under a **second**.

```
Running example.SimplePigTest
2014-05-25 13:33:56.306 java[1300:1003] Unable to load realm info from SCDynamicStore
Tests run: 1, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 0.968 sec
Test example.SimplePigTest FAILED
```

```
ries_limit =
 LIMIT queries_ordered $n;

RE queries_limit INTO 'output';
```

igUnit Example: Revised source script

```
a =
 LOAD 'input'
 AS (query:CHARARRAY);

ries_group =
 GROUP data
 BY query;

ries_count = FOREACH queries_group
 GENERATE group AS query,
     COUNT(data) AS total;
```

```
tsuite: example.SimplePigTest
ts run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 27.388 sec
---------- Standard Output ----------------
```

```
ries_limit =
 LIMIT queries_ordered $n;

RE queries_limit INTO 'output';
```

esting PIG UDFs

```
package myudfs;
import java.io.IOException;
import org.apache.pig.EvalFunc;
import org.apache.pig.data.Tuple;
import org.apache.pig.impl.util.WrappedIOException;

public class UPPER extends EvalFunc (String)
{
    public String exec(Tuple input) throws IOException {
        if (input == null || input.size() == 0)
            return null;
        try{
            String str = (String)input.get(0);
            return str.toUpperCase();
        }catch(Exception e){
            throw WrappedIOException.wrap("Caught exception processing input row ", e);
        }
    }
}
```

igUnit: Issues to be on the look out for

Only include sample data to perform the test

Some Pig features are missing:

- SHOW
- DUMP
- STORE

Review the logs!….even if there are no errors

Pig Job '/tmp' file contains useful additional information on failures ar errors

eetest

veloped by: Adam Kawa

tHub Project: https://github.com/kawaa/Beetest.git

etest provides a simple unit test capability on a given hadoop environment

equires a:

Setup script

Expected test outcome

Your query

eeTest – What we want to test

```
  SELECT artist, COUNT(*) AS cnt
        FROM streamed_songs
GROUP BY artist
ORDER BY cnt DESC
   LIMIT 2;
```

ource Data:

tist	Song	user	Date	Time
oldplay	Vivalavida	adam.kawa	1/1/13	21:20:10
oldplay	Vivalavida	natalia.stachura	1/1/13	21:22:41
asis	Wonderwall	adam.kawa	1/2/13	2:33:55
oldplay	Yelllow	adam.kawa	1/2/13	14:10:01
asis	Wonderwall	dog.tofi	1/2/13	22:17:51
erosmith	Crazy	natalia.stachura	1/2/13	23:48:31

Expected Results:

Artist	Count
Coldplay	3
Oasis	2

Hortonwo

eeTest – The test execution

st Setup:

- Define a directory to contain the test components
- Expected.txt – The Hive query output
- Input.tsv – Sample input
- Setup.hql – The test table DDL instructions
- Select.hql – Your test query (there is an option to change the file name)

```
root@sandbox examples]# ls artist-count
xpected.txt  input.tsv  select.hql  setup.hql
root@sandbox examples]#
```

mand Invocation:

```
ot@sandbox examples]#
ot@sandbox examples]# ./run-test.sh artist-count local-config/
```

eeTest – Interpreting test results

ccess:

```
26, 2014 1:44:46 PM com.spotify.beetest.Utils runCommand
: Time taken: 17.441 seconds
26, 2014 1:44:46 PM com.spotify.beetest.TestQueryExecutor run
: Asserting: artist-count/expected.txt and /tmp/beetest-test-225777919-output_225777919/000000_0
t@sandbox examples]#
```

lure:

```
: Asserting: artist-count/expected.txt and /tmp/beetest-test-260960455-output_260960455/000000_0
ption in thread "main" junitx.framework.ComparisonFailure: Output does not match Line [3] expecte
asis   3> but was:<Oasis       2> ['2']
    at junitx.framework.Assert.assertEquals(Assert.java:120)
    at junitx.framework.FileAssert.assertEquals(FileAssert.java:183)
```


Hortonwo

etting started with your testing initiative

Start Simple

Materiality rule: Focus on Highest value tests which are the easiest to implement

Keep things “light and fast”

Create an automated environment (Jenkins, AntHill Pro, etc) for production used Jobs

Keep track of results

uestions?

Mark Johnson
Regional Director Consulting
Hortonworks

mjohnson@hortonworks.com
markfjohnson@gmail.com

Twitter: markfjohnson

Hortonworks is Hiring!

ortonworks Data Platform

e do Hadoop

Hortonwo