Production殊死戦

在Production遇到的困難？

建立Hive Table沒辦法Partition

建立Hive Table後沒辦法SELECT

LOG 爆掉！！！！！！！

要如何在Production上活下來？

Production 環境變數

```
# Spark
export SPARK_HOME=/etc/spark-2.1.0-bin-hadoop2.6/
export PATH=$PATH:$SPARK_HOME/bin
export PYTHONPATH=$SPARK_HOME/python/lib/pyspark.zip:$PYTHONPATH
export PYTHONPATH=$SPARK_HOME/python/lib/py4j-0.10.3-src.zip:$PYTHONPATH
export HADOOP_CONF_DIR=/source/hadoop/conf
```

啟動 jupyter

HADOOP_CONF_DIR=/source/hadoop/conf

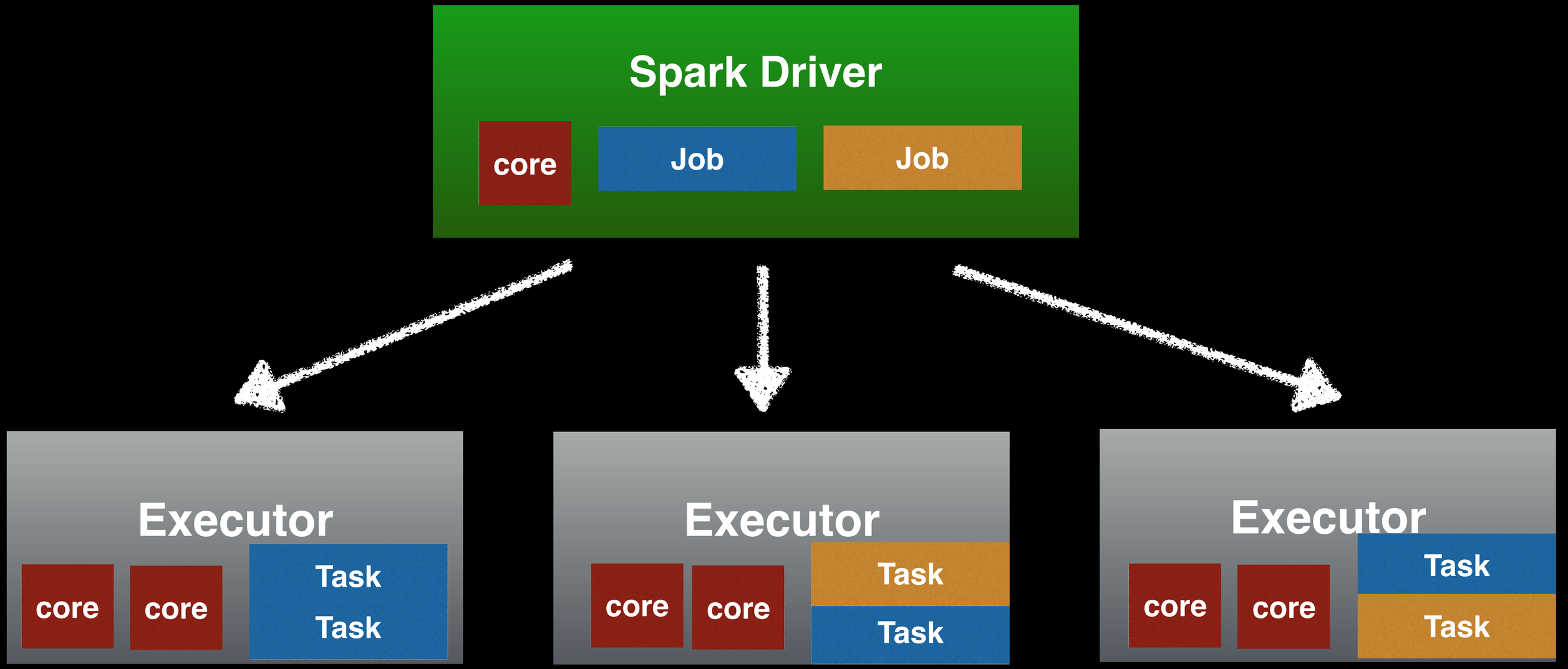
SPARK_HOME=/etc/spark-2.1.0-bin-hadoop2.6

PYSPARK_DRIVER_PYTHON=jupyter

PYSPARK_DRIVER_PYTHON_OPTS="notebook --ip 88.8.146.34 --port 9999"

/etc/spark-2.1.0-bin-hadoop2.6/bin/pyspark --name pyspark-roger --master yarn --deploy-mode client --driver-cores 1 --driver-memory 2g --executor-memory 8g --executor-cores 4 --num-executors 16

資源就那麼多，要如何調配？


Spark Driver
core
Job
Job
Executor
core
core
Task
Task
Executor
core
core
Task
Task
Executor
core
core
Task
Task

Spark-Submit Options

Driver：向RM進行資源的申請、任務的分配和監控等；當Executor部分運行完畢後，Driver負責將SparkContext關閉。

Executor：負責運行Task，並且負責將數據存在內存或者磁碟上。

Executor-cores：CPU core同一時間只能執行一個Task。

Job：包含多個Task組成的並行計算，往往由Spark Action催生，一個JOB包含多個RDD及作用於相應RDD上的各種Operation。

Stage：每個Job會被拆分很多組Task，每組任務被稱為Stage，也可稱TaskSet，一個作業分為多個階段。

Task： 被送到某個Executor上的工作任務。

Spark-Submit Options

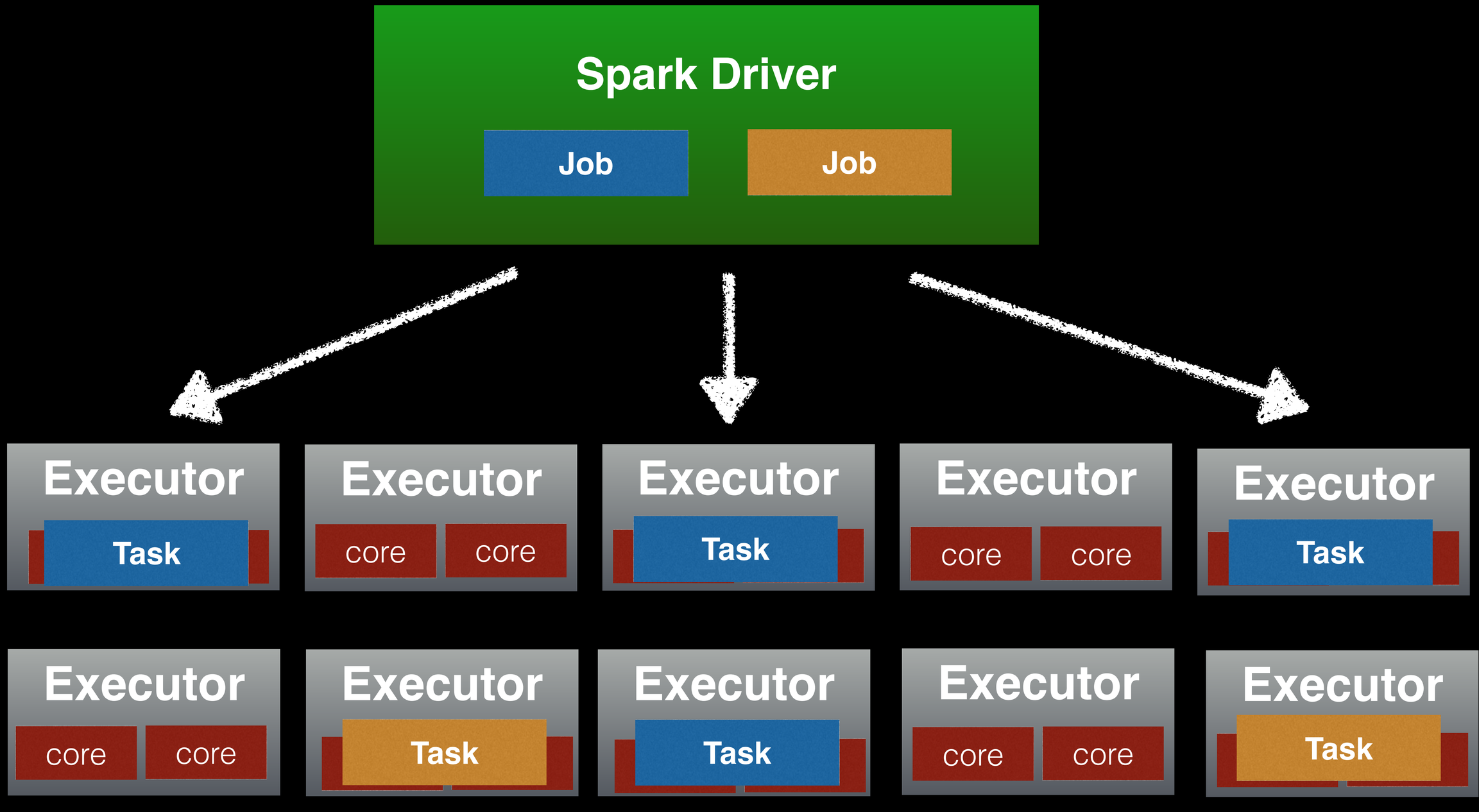
--master yarn (local)
--deploy mode client (cluster)
--driver-memory 2g
--driver-cores 1
--num-executors 16
--executor-cores 4
--executor-memory 2g
--py-files [‘hello.py’,’rc.py’]

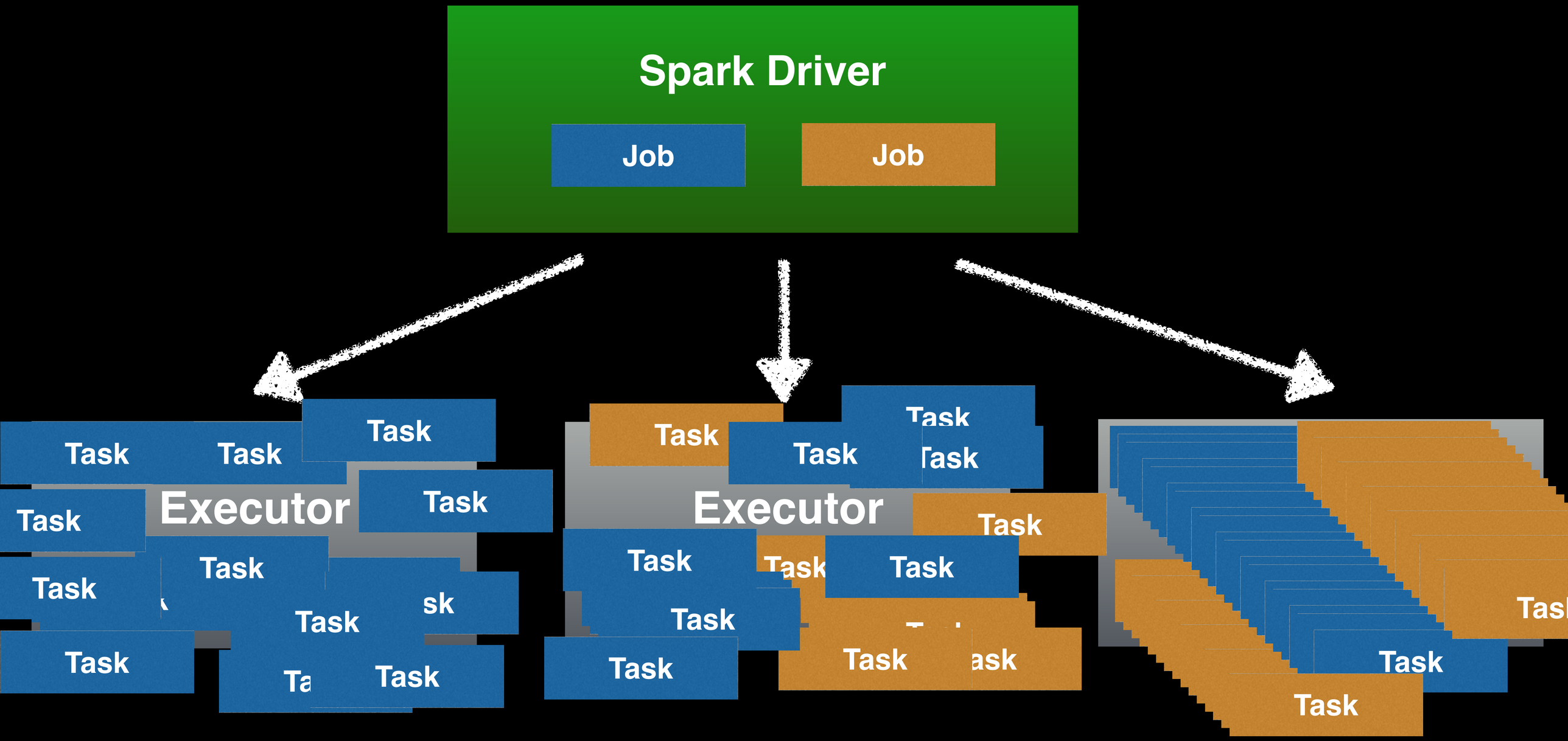
Command-line: spark-submit

Set Spark Configuration

```
hc.setConf("hive.exec.dynamic.partition.mode", “nonstrict")

hc.setConf("spark.sql.parquet.compression.codec", "uncompressed")

hc.setConf(“spark.default.parallelism”,”2048”)

hc.setConf("spark.sql.shuffle.partitions", “4”)
```

Spark-Submit後會發生什麼事？


Spark Driver
Job
Job
Executor
Task
Executor
core
core
Executor
Task
Executor
core
core
Executor
Task
Executor
core
core
Executor
Task
Executor
Task
Executor
core
core
Executor
core
core
Executor
Task


Spark Driver
Job
Job
Executor
Executor
Task

最常發生的問題是？

Out of Memory!!!!!!!!

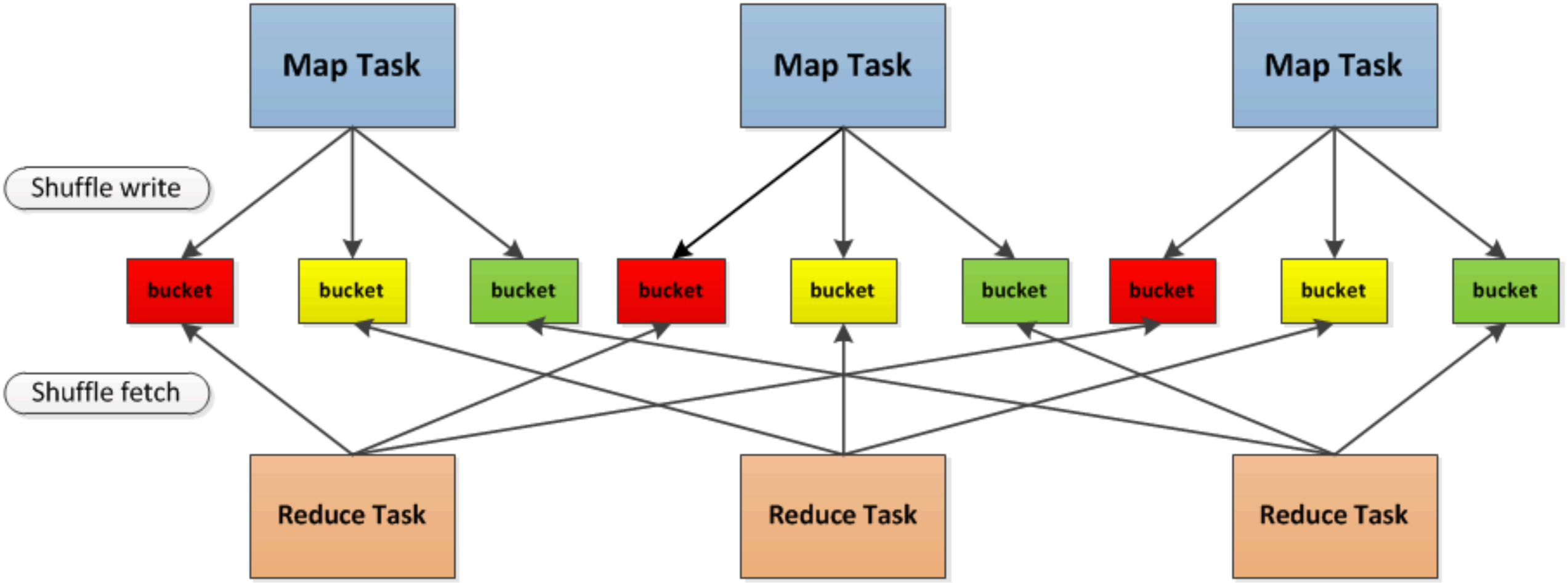
無限加大Memory吧！！！

不可能！!!!

無限加大Memory吧！！！

Shuffle

MapReduce框架中，Shuffle是連結Map和Reduce之間的橋樑，Map的輸出要用到Reduce中必須經過Shuffle這個環節，Shuffle的性能高低直接影響了整個程序性能。

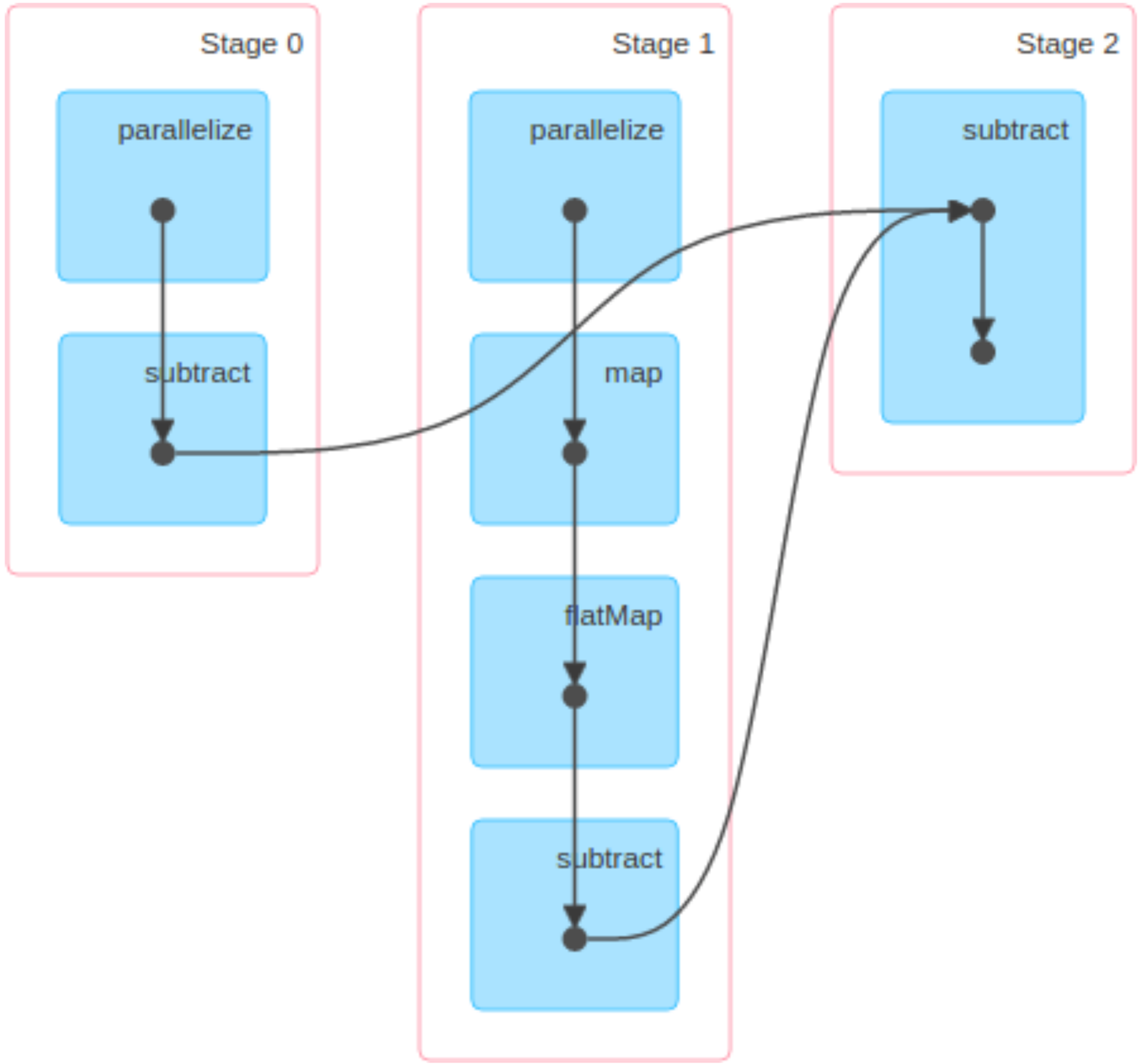


Shuffle就是各種需要重分區的算子之下的一個對數據重新组合的過程

Repartition

尋找質數

- 目標找出2到2000000所有質數
- 找出所有非質數
- 從2開始找出所有倍數，再找3所有倍數以此類推……



Stage 0

Tasks

Index ▲	ID	Attempt	Status	Locality Level	Executor ID / Host	Launch Time	Duration	GC Time	Write Time	Shuffle Write Size / Records	Errors
0	0	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.5 s	34 ms	12 ms	1008.3 KB / 249999	
1	1	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.6 s	34 ms	9 ms	1008.9 KB / 250000	
2	2	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.5 s	34 ms	12 ms	1008.8 KB / 250000	
3	3	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.5 s	34 ms	15 ms	1008.6 KB / 250000	
4	4	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.5 s	34 ms	10 ms	1008.9 KB / 250000	
5	5	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.6 s	34 ms	14 ms	1008.3 KB / 250000	
6	6	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.5 s	34 ms	13 ms	1008.9 KB / 250000	
7	7	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:29	0.5 s	34 ms	12 ms	1008.6 KB / 250000	

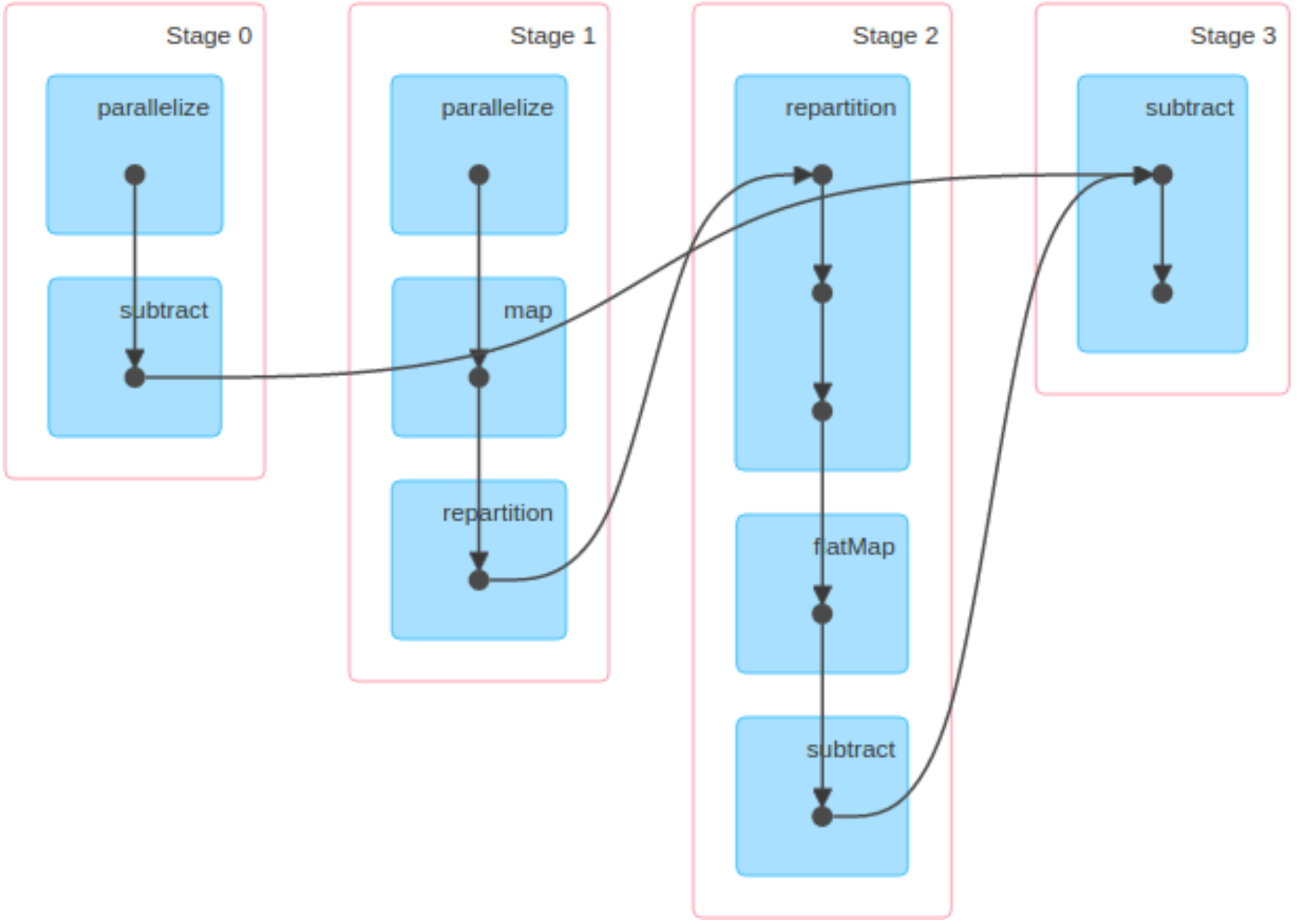
Stage 1

Tasks

Index ▲	ID	Attempt	Status	Locality Level	Executor ID / Host	Launch Time	Duration	GC Time	Write Time	Shuffle Write Size / Records	Errors
0	8	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	14 s	0.2 s	0.2 s	106.3 MB / 23640584	
1	9	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	1 s	0.1 s	22 ms	4.6 MB / 1019047	
2	10	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	0.8 s	0.1 s	5 ms	1683.9 KB / 416666	
3	11	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	0.6 s	0.1 s	4 ms	1008.9 KB / 250000	
4	12	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	0.4 s	96 ms		0.0 B / 0	
5	13	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	0.4 s	96 ms		0.0 B / 0	
6	14	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	0.3 s	96 ms		0.0 B / 0	
7	15	0	SUCCESS	PROCESS_LOCAL	driver / localhost	2016/05/05 14:42:30	0.2 s	90 ms		0.0 B / 0	

Repartition

- Spark Partition : 數據中的基本組成單位
- 運算過程中數據量時大時小，選擇合適的partition數量關係重大，如果太多partition就導致有很多小任務和空任務產生；如果太少則導致運算資源沒法充分利用，必要時候可以使用repartition來調整，不過它也不是沒有代價的，其中一個最主要代價就是shuffle。
- shuffle :是兩個 stage 之間的數據傳輸過程。


Stage 0
parallelize
subtract
Stage 1
parallelize
map
repartition
Stage 2
repartition
flatMap
subtract
Stage 3
subtract

Tasks

Index ▲	ID	Attempt	Status	Locality Level	Executor ID / Host	Launch Time	Duration	GC Time	Shuffle Read Size / Records	Write Time	Shuffle Write Size / Records	Errors
0	16	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	5 s	0.2 s	2.7 MB / 250000	42 ms	14.2 MB / 3242491	
1	17	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	5 s	0.2 s	2.7 MB / 250000	44 ms	13.9 MB / 3114793	
2	18	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	5 s	0.2 s	2.7 MB / 250000	36 ms	13.5 MB / 3023100	
3	19	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	5 s	0.2 s	2.7 MB / 250000	38 ms	13.3 MB / 2952632	
4	20	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	5 s	0.2 s	2.7 MB / 250000	56 ms	12.7 MB / 2895958	
5	21	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	4 s	0.2 s	2.7 MB / 249999	53 ms	12.9 MB / 2848739	
6	22	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	5 s	0.2 s	2.7 MB / 250000	41 ms	16.6 MB / 3808486	
7	23	0	SUCCESS	NODE_LOCAL	driver / localhost	2016/05/05 14:50:17	5 s	0.2 s	2.7 MB / 250000	36 ms	15.2 MB / 3440098	

Dashboard

Devices

IP Address

Session

Login Time

individual

Cookie

Pages

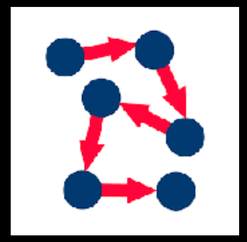
Devices

IP Address

Duration

Session

Login Time



Page Sequence

individual

Cookie

domains

Pages

Devices

IP Address

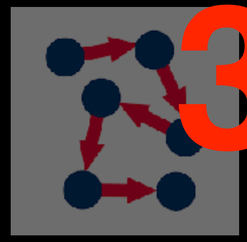
Duration

一天約200萬筆資料

Session

Login Time

一個月不含app約3000萬筆資料



Page Sequence

individual

Cookie

domains

困難點在哪？

日期區間： 2017-01-19 ~ 2017-01-23

不重複訪客數	瀏覽量	辨識率	新訪客佔比	平均瀏覽頁數	平均停留時間
617,703	4,787,002	41.06%	66.59%	4.25	00:16:53

全選						
Web-PC	173,353	865,614	39.58%	71.45%	2.85	00:43:09
Web-Mobile	143,156	617,538	11.05%	80.58%	2.78	00:10:24
MyBank-PC	178,533	1,349,217	83.27%	63.02%	4.61	00:54:33
MyBank-Mobile	207,952	1,533,671	44.95%	55.12%	3.86	00:21:03
B2B	20,338	402,055	58.47%	44.23%	6.67	00:22:59
KOKO	4537	18,907	4.29%	87.23%	3.42	00:56:49

總覽

線上申辦流程轉換效益

效益分析

流程比較

回訪率分析

日期區間： 2017-01-19 ~

全選

Web-PC

Web-Mobile

MyBank-PC

MyBank-Mobile

B2B

KOKO

自測覽頁數 4.25

平均停留時間 00:16:53

00:43:09

00:10:24

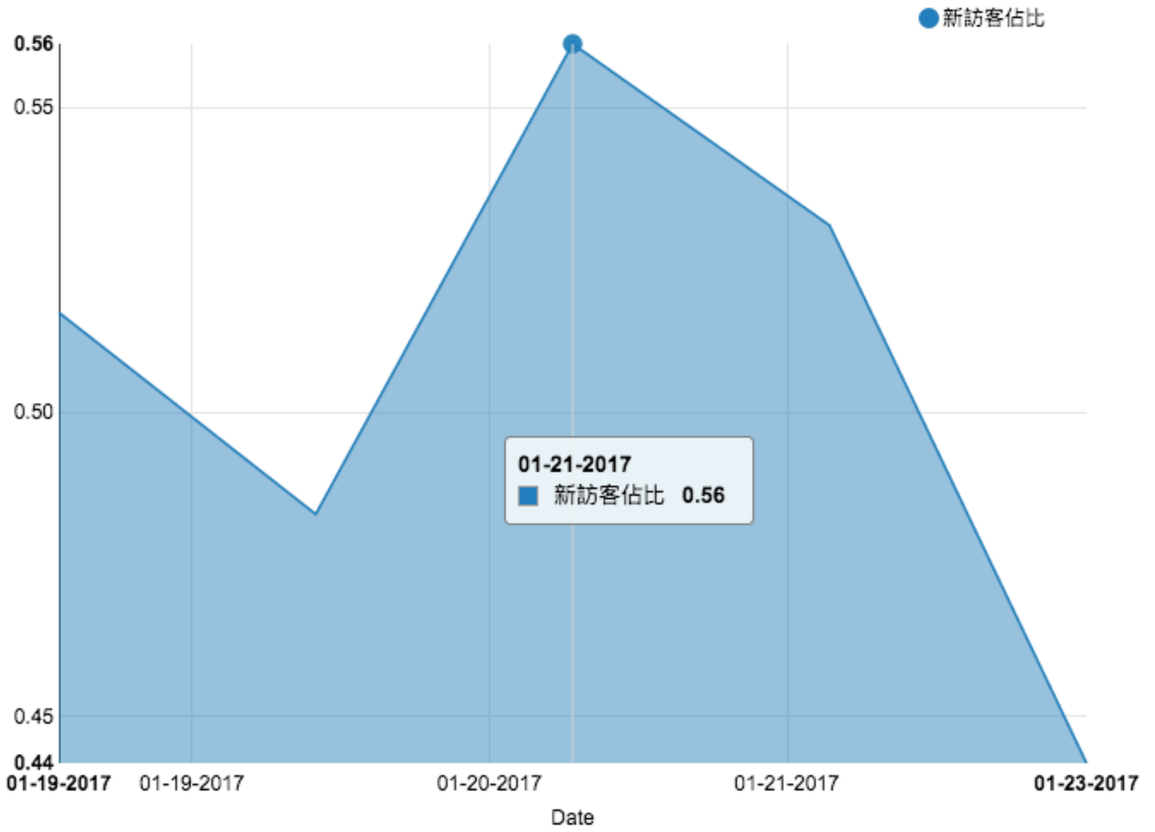
00:54:33

00:21:03

00:22:59

00:56:49

趨勢變化



總覽

線上申辦流程轉換效益

效益分析

流程比較

回訪率分析

日期區間： 2017-01-19 ~

全選

Web-PC

Web-Mobile

MyBank-PC

MyBank-Mobile

B2B

KOKO

平均停留時間

4.25

00:16:53

00:43:09

00:10:24

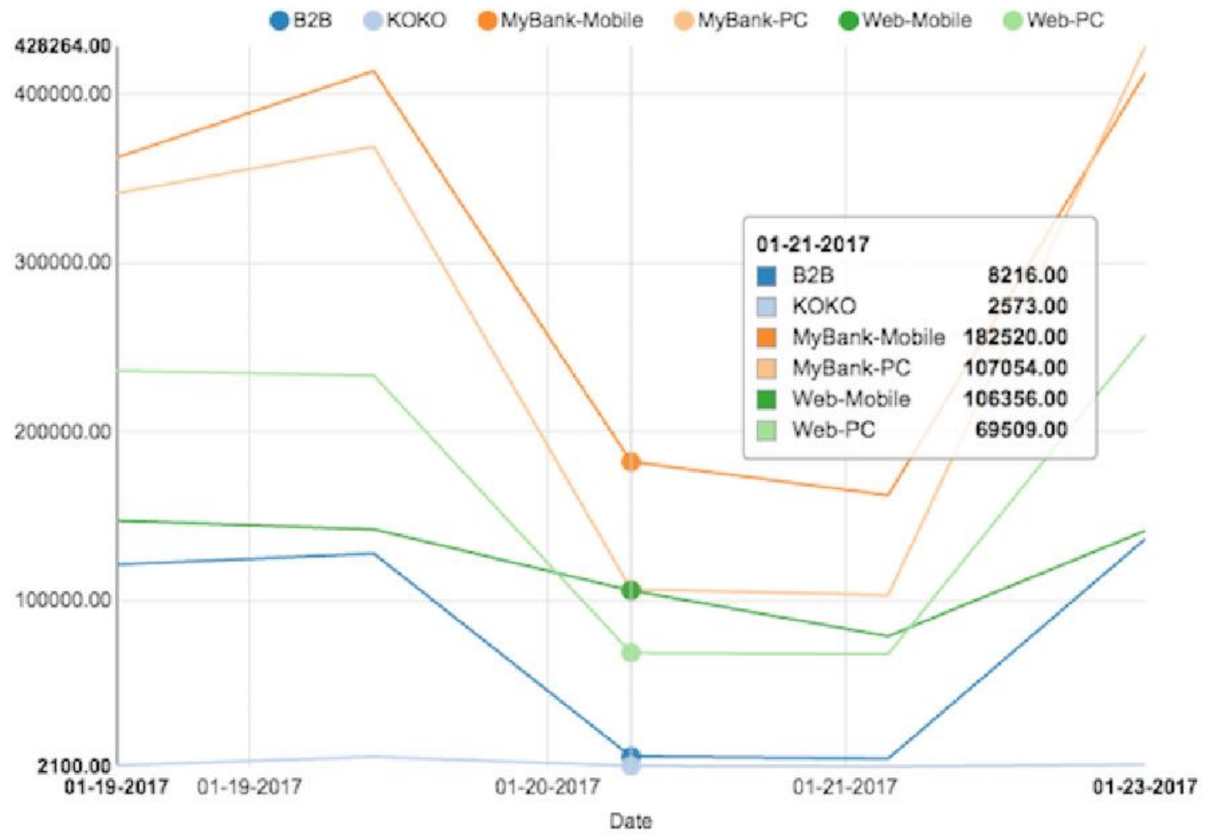
00:54:33

00:21:03

00:22:59

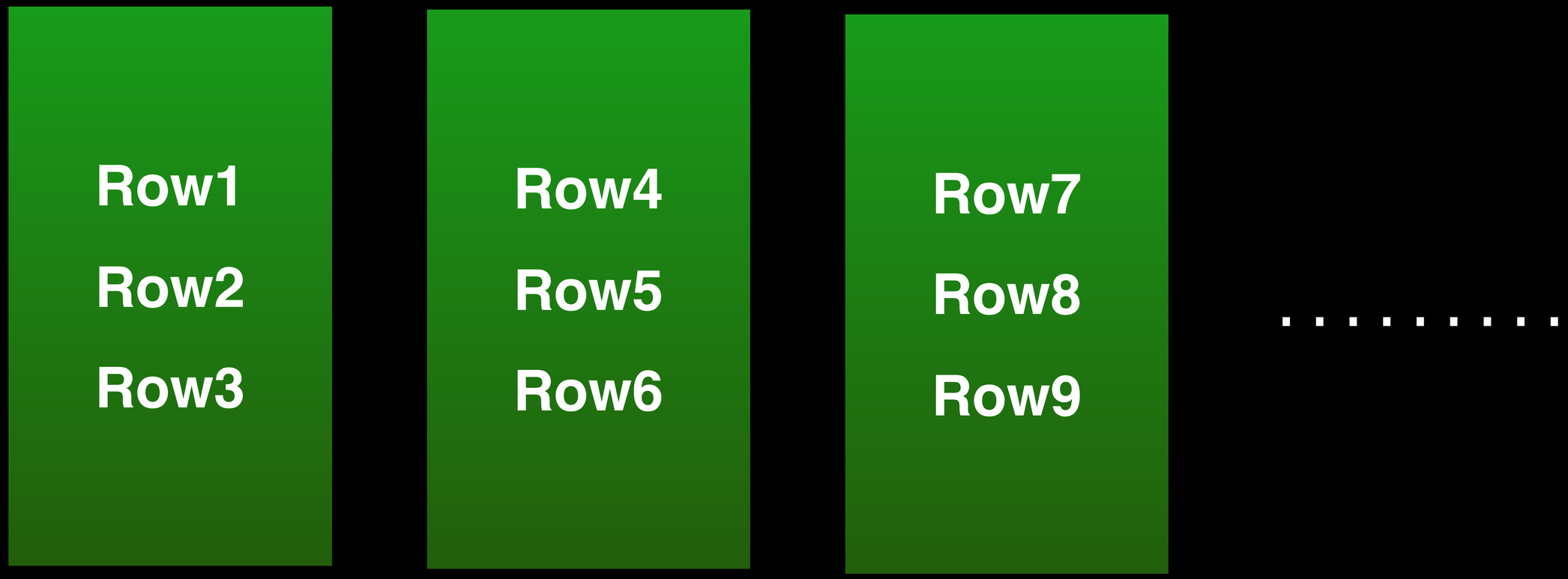
00:56:49

趨勢變化

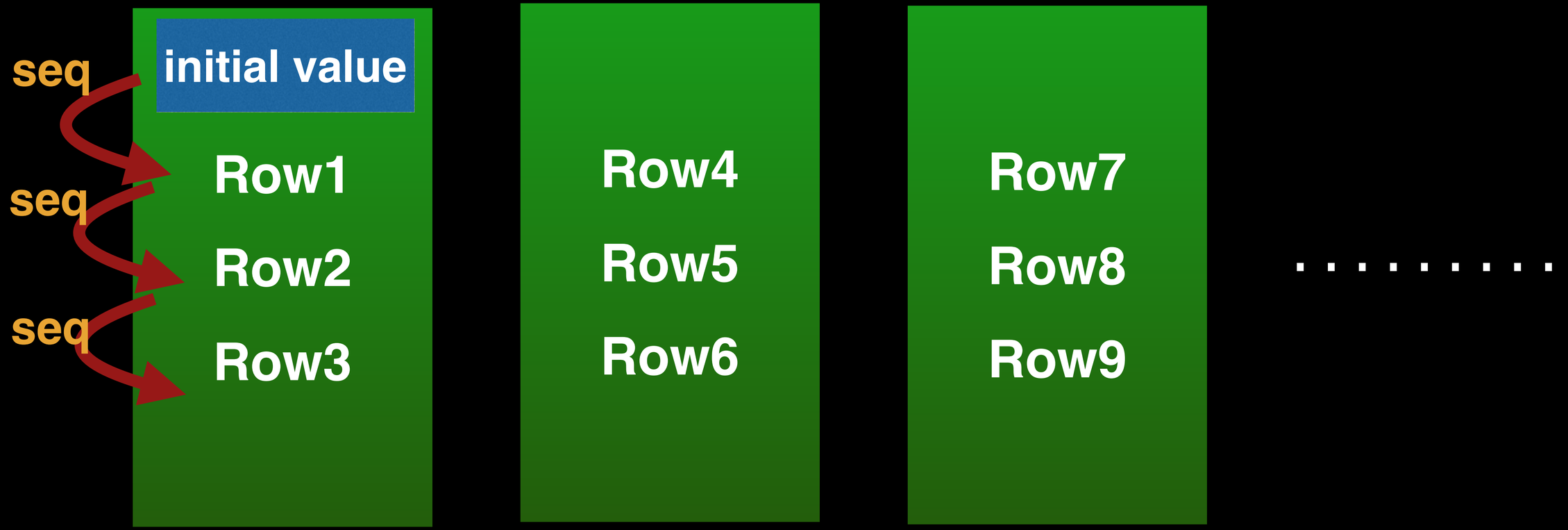


AggregateByKey

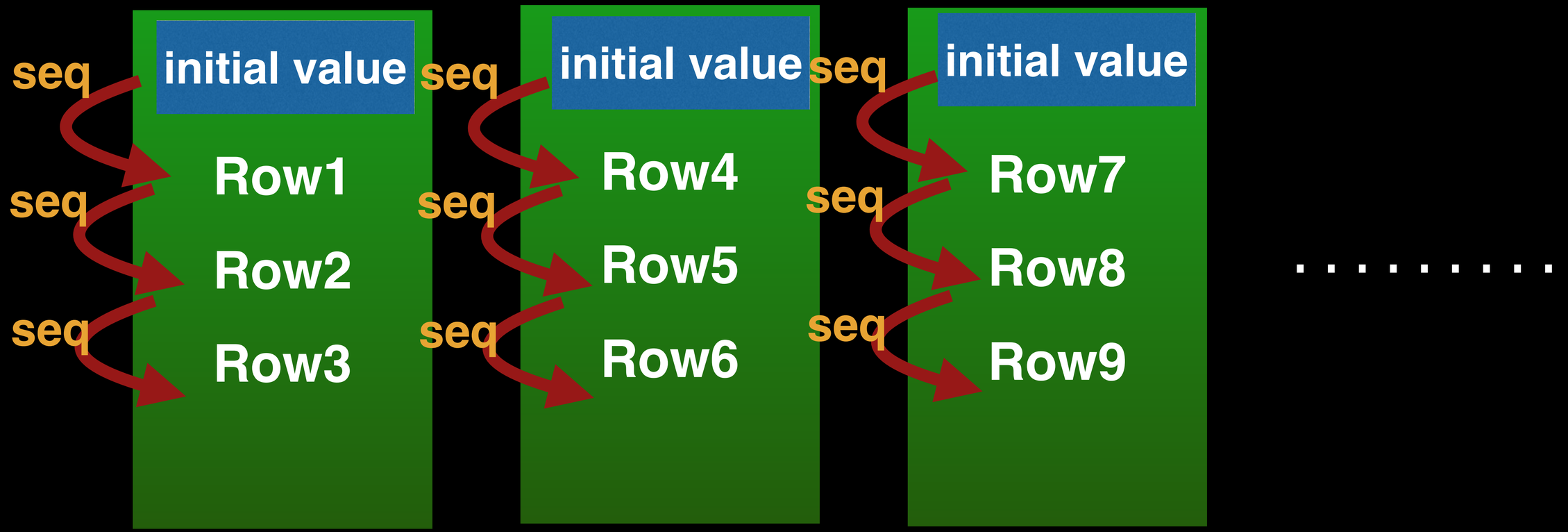
AggregateByKey
(initial value, sequence, combine, numpartitions)



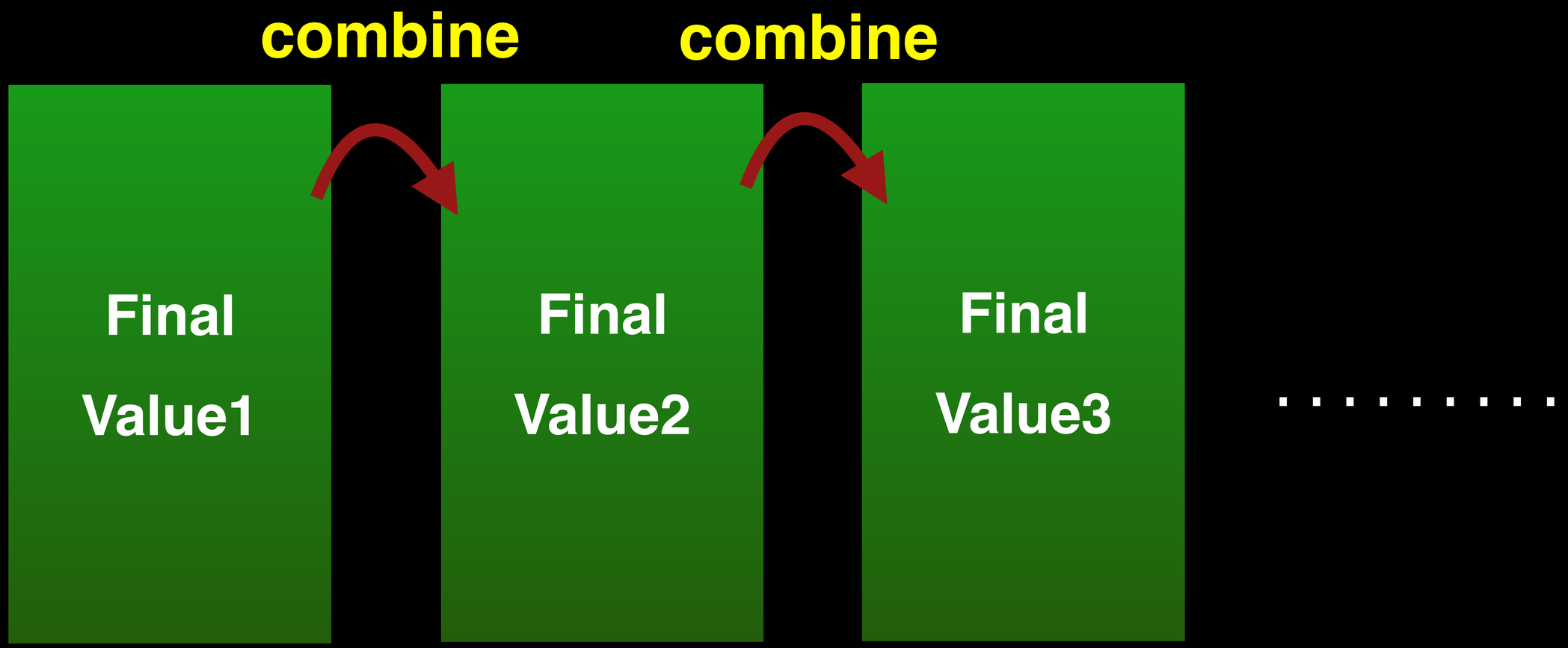
AggregateByKey
(initial value, sequence, combine, numpartitions)



AggregateByKey
(initial value, sequence, combine, numpartitions)



AggregateByKey
(initial value, sequence, combine, numpartitions)

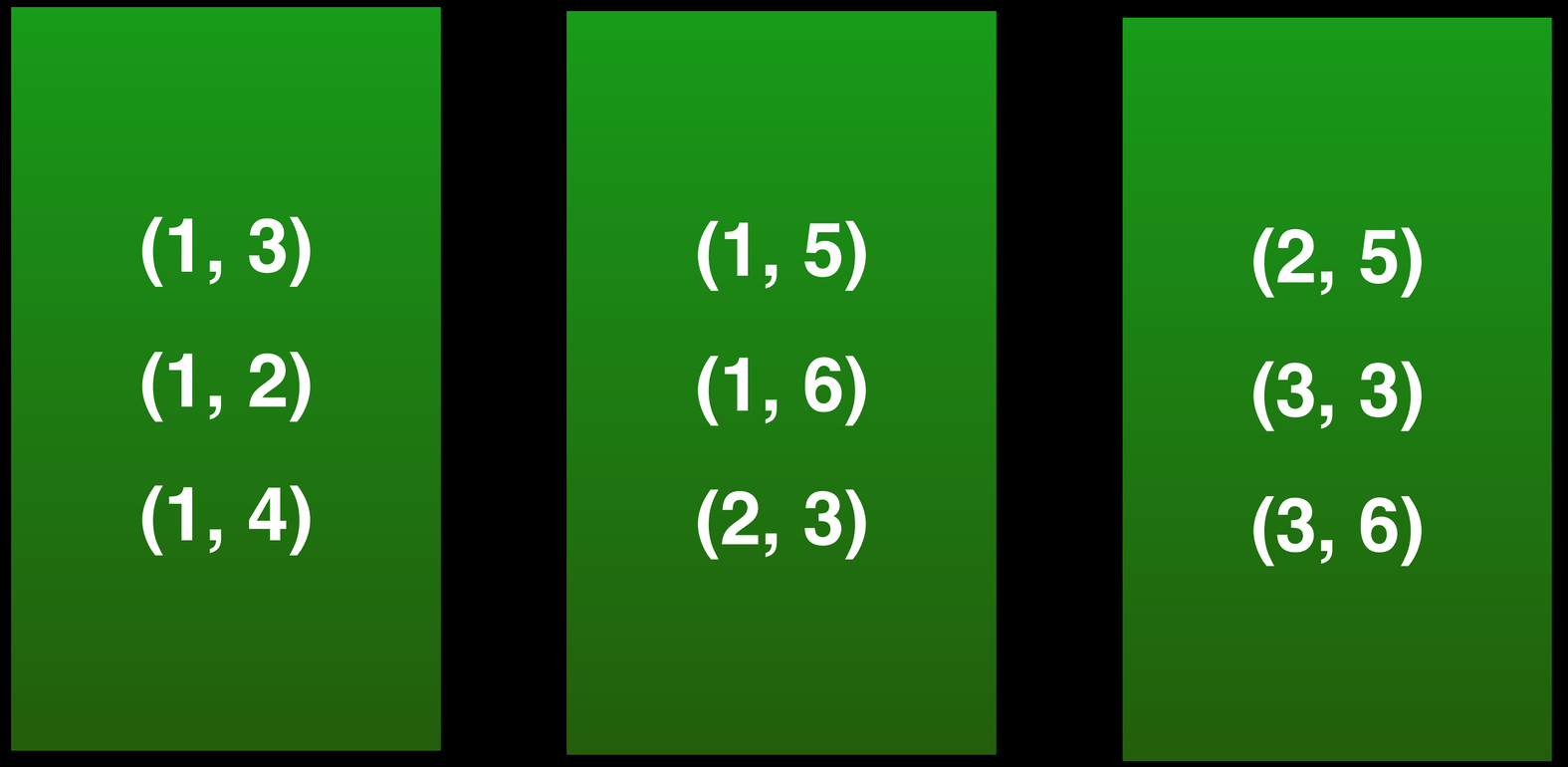


Try AggregateByKey

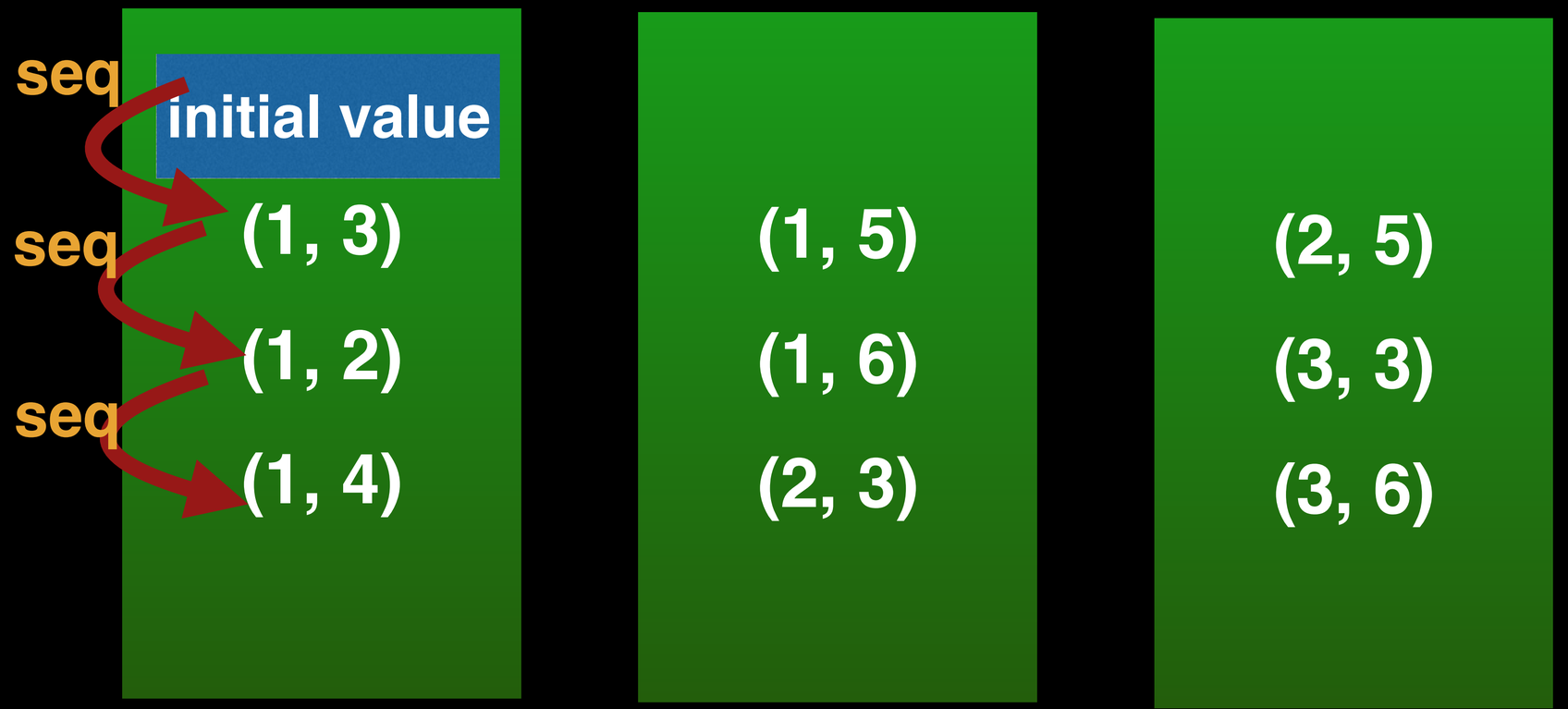
```
def seq(a,b):
    return a*b

def combine(a,b):
    return a+b
```

```
data = sc.parallelize([(1,3),(1,2),(1,4),\
                       (1,5),(1,6),(2,3),\
                       (2,5),(3,3),(3,6)],3)
data.aggregateByKey(2,seq,combine,10).collect()
```

AggregateByKey
(2, sequence, combine, 4)

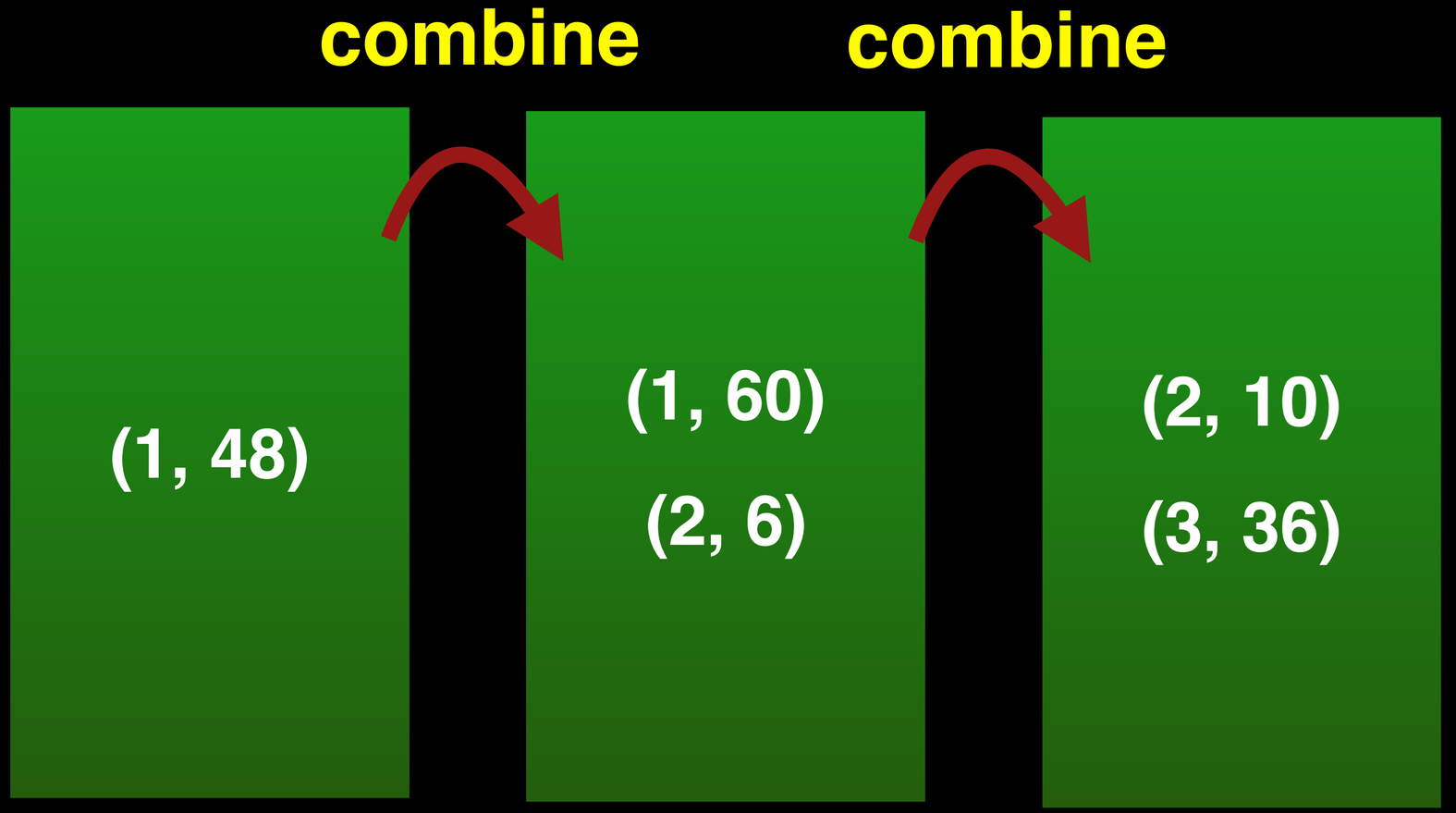


AggregateByKey
(2, sequence, combine, 4)



AggregateByKey
(2, sequence, combine, 4)

AggregateByKey
(2, sequence, combine, 4)



AggregateByKey
(2, sequence, combine, 4)

Hands On

- Reduce Task的內存使用。在某些情況下reduce task特別消耗內存，比如當shuffle出現的時候，比如sortByKey、groupByKey、reduceByKey和join等，要在內存裡面建立一個巨大的hash table。其中一個解決辦法是增大level of parallelism，這樣每個task的輸入規模就相應減小。另外，注意shuffle的內存上限設置，有時候有足夠的內存，但是shuffle內存不夠的話，性能也是上不去的。我們在有大量數據join等操作的時候，shuffle的內存上限經常配置到executor的50%。

Web-PC

MyBank-PC

Web-Mobile

Domains

MyBank-Mobile

B2B

KOKO