Spark Core

Part 2

Justin Pihony

@JustinPihony | justin-pihony.blogspot.com

Module Overview

- **Core API**
  - Appify
  - RDD
  - Transforming Data
  - Action!
  - KeyValue
  - Cache
  - Accumulator
  - Java

RDD Implicits

doubleRDDToDoubleRDDFunctions(rdd: RDD[Double]): DoubleRDDFunctions

numericRDDToDoubleRDDFunctions[T](rdd: RDD[T]): DoubleRDDFunctions

rddToAsyncRDDActions[T](rdd: RDD[T]): AsyncRDDActions[T]

rddToOrderedRDDFunctions[K,V](rdd: RDD[(K,V)]): OrderedRDDFunctions

rddToPairRDDFunctions[K,V](rdd: RDD[(K,V)]): PairRDDFunctions[K,V]

rddToSequenceFileRDDFunctions[K,V](rdd: RDD[(K,V)]): SequenceFileRDDFunctions[K,V]

RDD Implicits

doubleRDDToDoubleRDDFunctions(rdd: RDD[Double]): DoubleRDDFunctions

numericRDDToDoubleRDDFunctions[T](rdd: RDD[T]): DoubleRDDFunctions

rddToAsyncRDDActions[T](rdd: RDD[T]): AsyncRDDActions[T]

rddToOrderedRDDFunctions[K,V](rdd: RDD[(K,V)]): OrderedRDDFunctions

rddToPairRDDFunctions[K,V](rdd: RDD[(K,V)]): PairRDDFunctions[K,V]

rddToSequenceFileRDDFunctions[K,V](rdd: RDD[(K,V)]): SequenceFileRDDFunctions[K,V]

Pairs

A	One
A	One(1)
C	Three
D	Four
D	Four(1)
D	Four(2)
G	Seven

Pairs

A	One
A	One(1)
C	Three
D	Four
D	Four(1)
D	Four(2)
G	Seven

Pairs

Node 1

A	One
A	One(1)
C	Three

Node 2

D	Four
D	Four(1)
D	Four(2)

Node 3

G	Seven

Pairs

Node 1

A	One
A	One(1)
C	Three

Node 2

D	Four
D	Four(1)
D	Four(2)

Node 3

G	Seven

Pairs

Node 1

A	One
A	One(1)
C	Three

Node 2

D	Four
D	Four(1)
D	Four(2)

Node 3

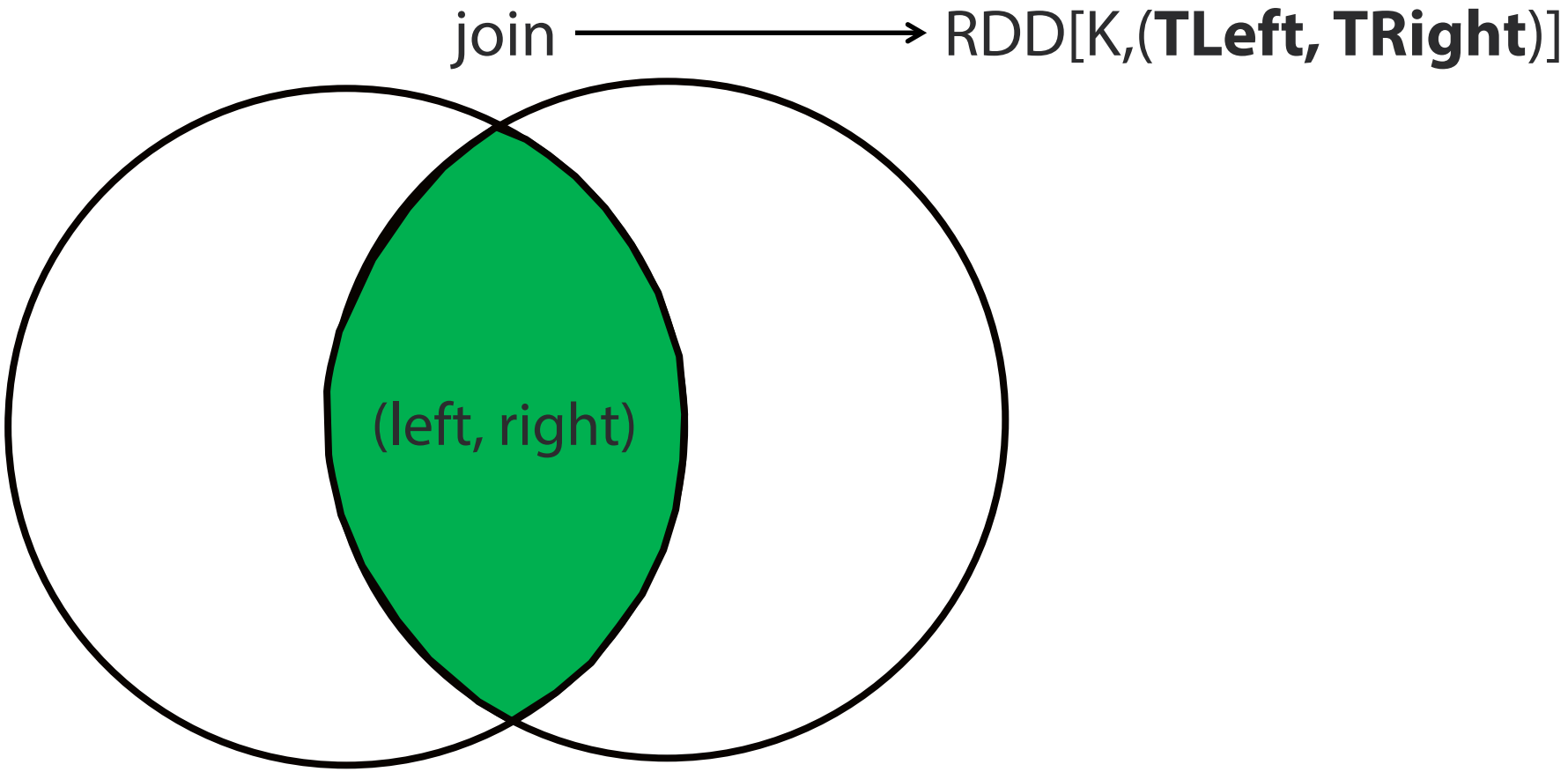
G	Seven

Pair Methods

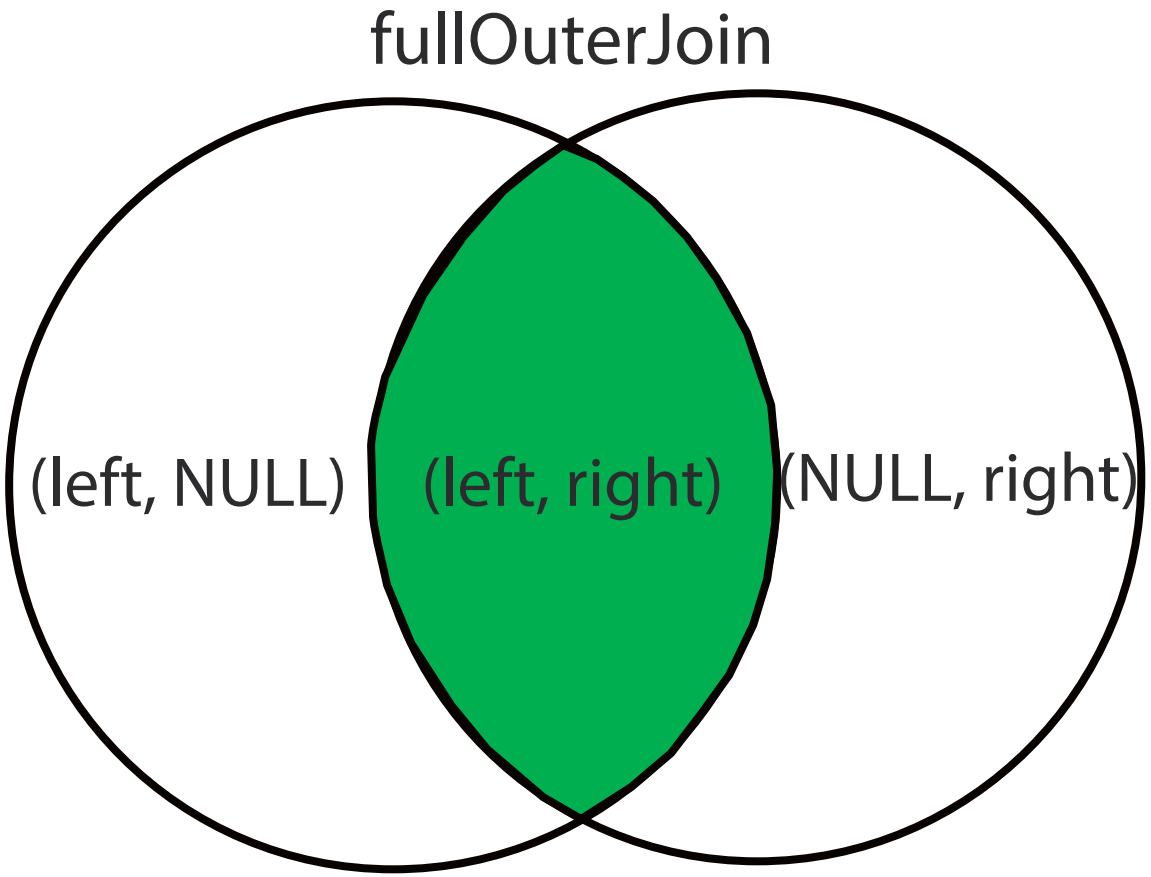
- collectAsMap
  - keys/values/lookup
- mapValues
- flatMapValues
- reduceByKey
  - …locally
- foldByKey
- aggregateByKey
- combineByKey

- groupByKey
- countByKey
- countApproxDistinctByKey
- sampleByKey
- subtractByKey
- sortByKey
  - OrderedRDDFunctions

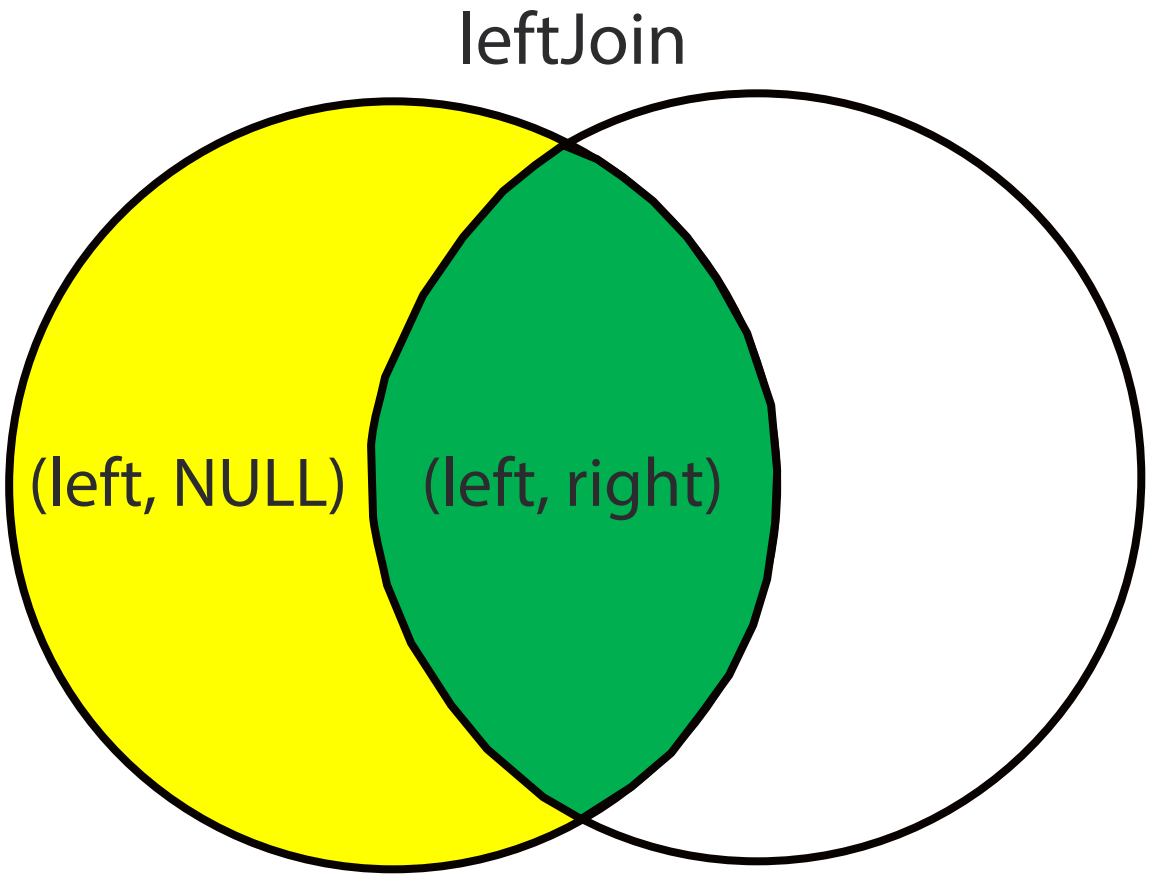
SQL-Like Pairings



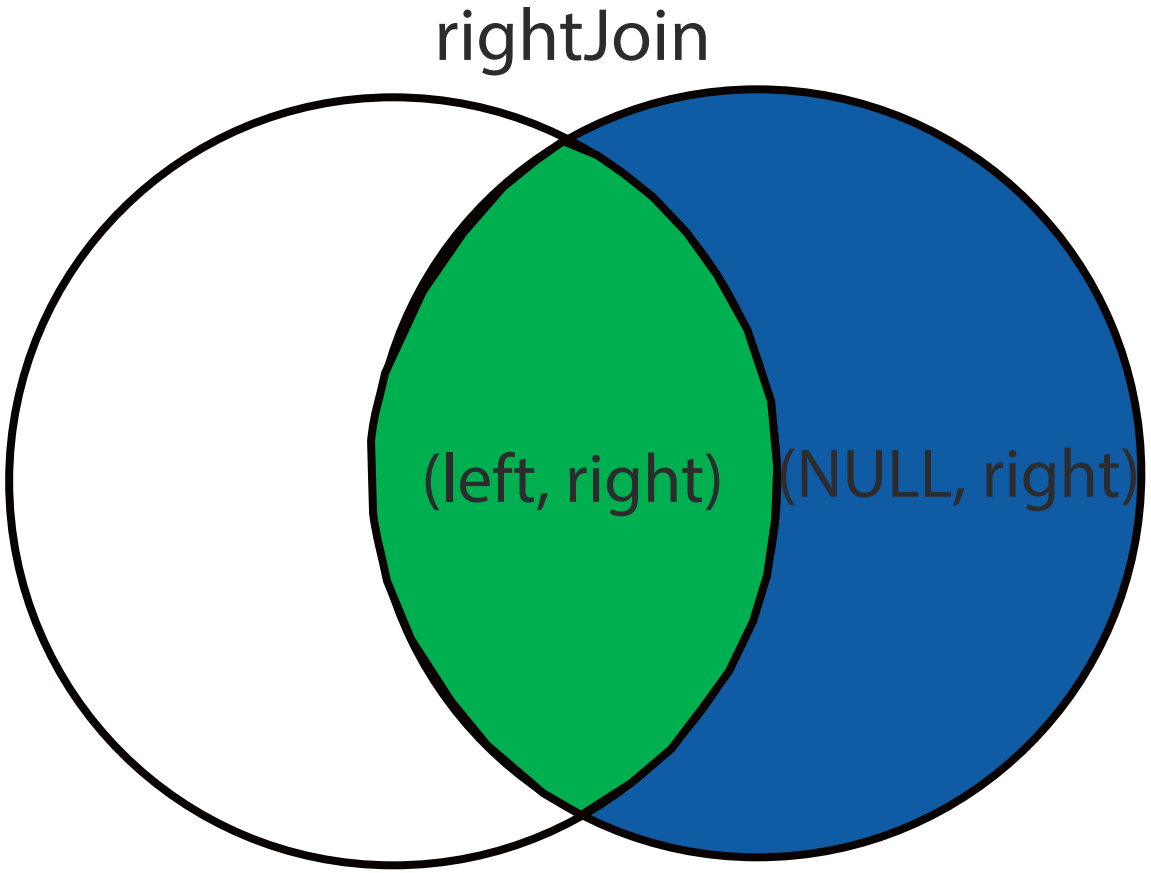
SQL-Like Pairings



SQL-Like Pairings

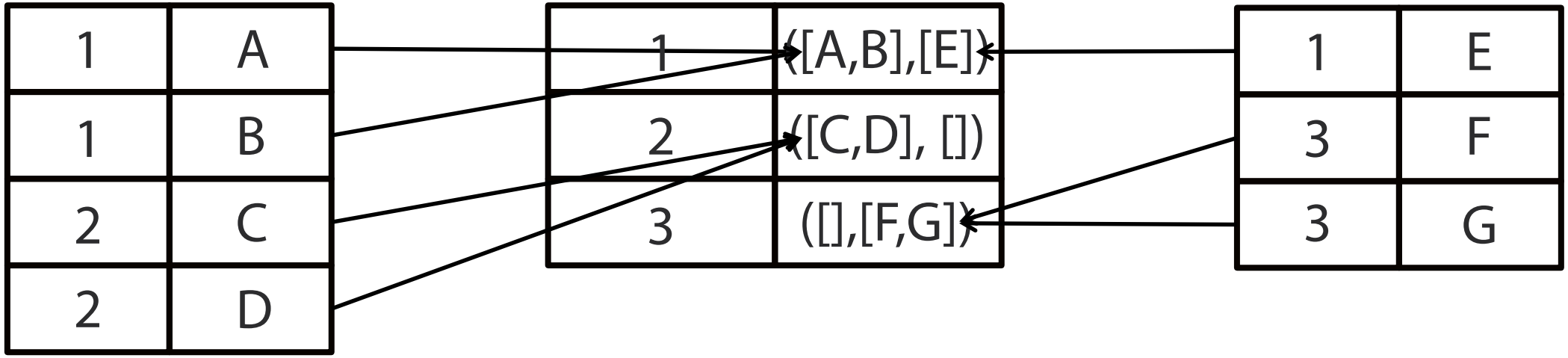


SQL-Like Pairings



Pair Methods

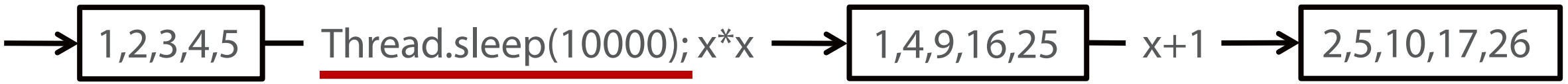
cogroup/groupWith



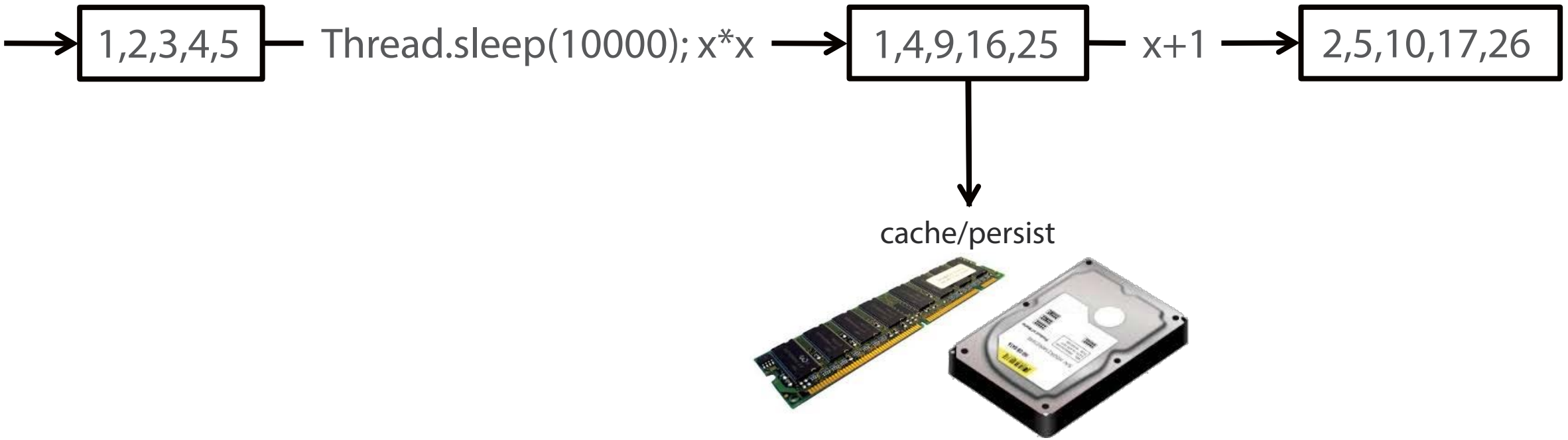
Pair Saving

- saveAs(NewAPI)HadoopFile
  - path
  - keyClass
  - valueClass
  - outputFormatClass
- saveAs(NewAPI)HadoopDataSet
  - conf
- saveAsSequenceFile
  - saveAsHadoopFile(path, keyClass, valueClass, **SequenceFileOutputFormat**)

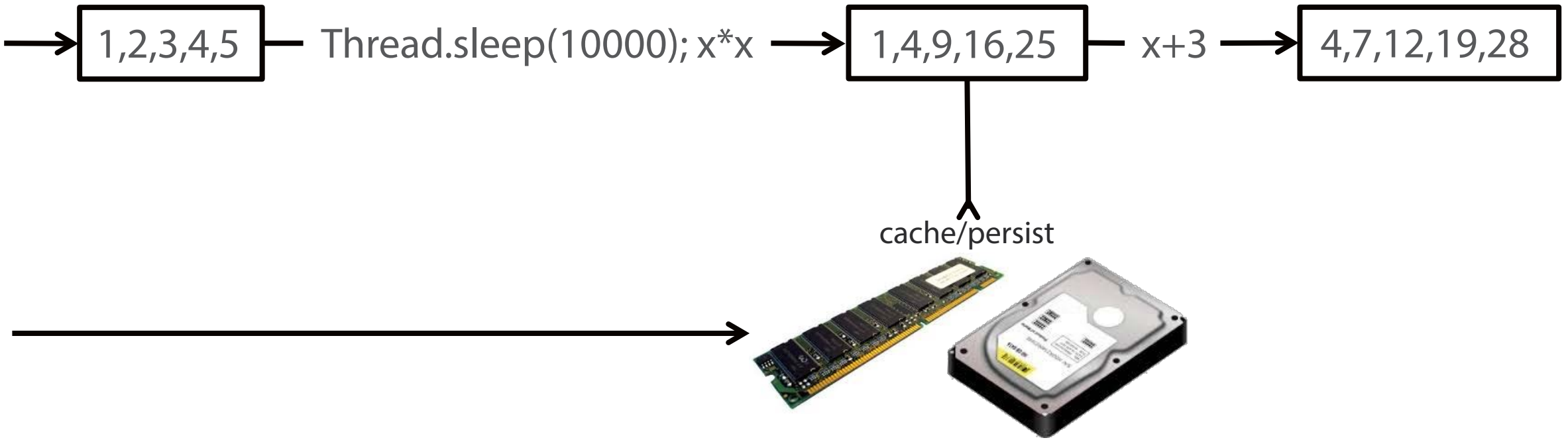
Cache



Cache



Cache



Cache

- cache/persist
  - org.apache.spark.storage.**StorageLevel.MEMORY_ONLY**
- persist(newLevel: StorageLevel)
  - MEMORY_ONLY
  - MEMORY_AND_DISK
  - DISK_ONLY
  - MEMORY_ONLY_SER
  - MEMORY_AND_DISK_SER
  - …_2
  - OFF_HEAP
- unpersist(blocking: Boolean = true)

Accumulator

```
val accumulator = sc.accumulator(0, “Accumulator Name”)
```

: Accumulator[Int]

```
rdd.foreach(x => {
  doSomethingWith(x)
  accumulator += 1
})
```

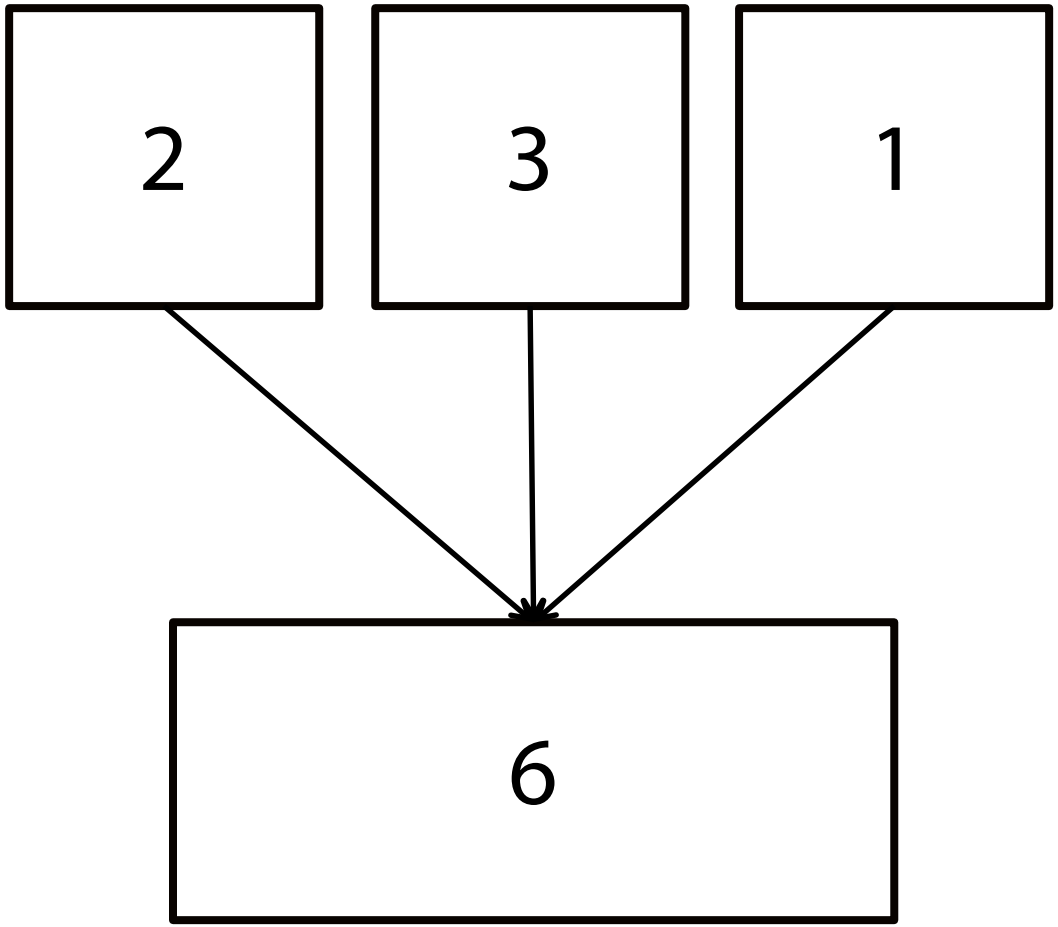
Action Methods

Worker Task

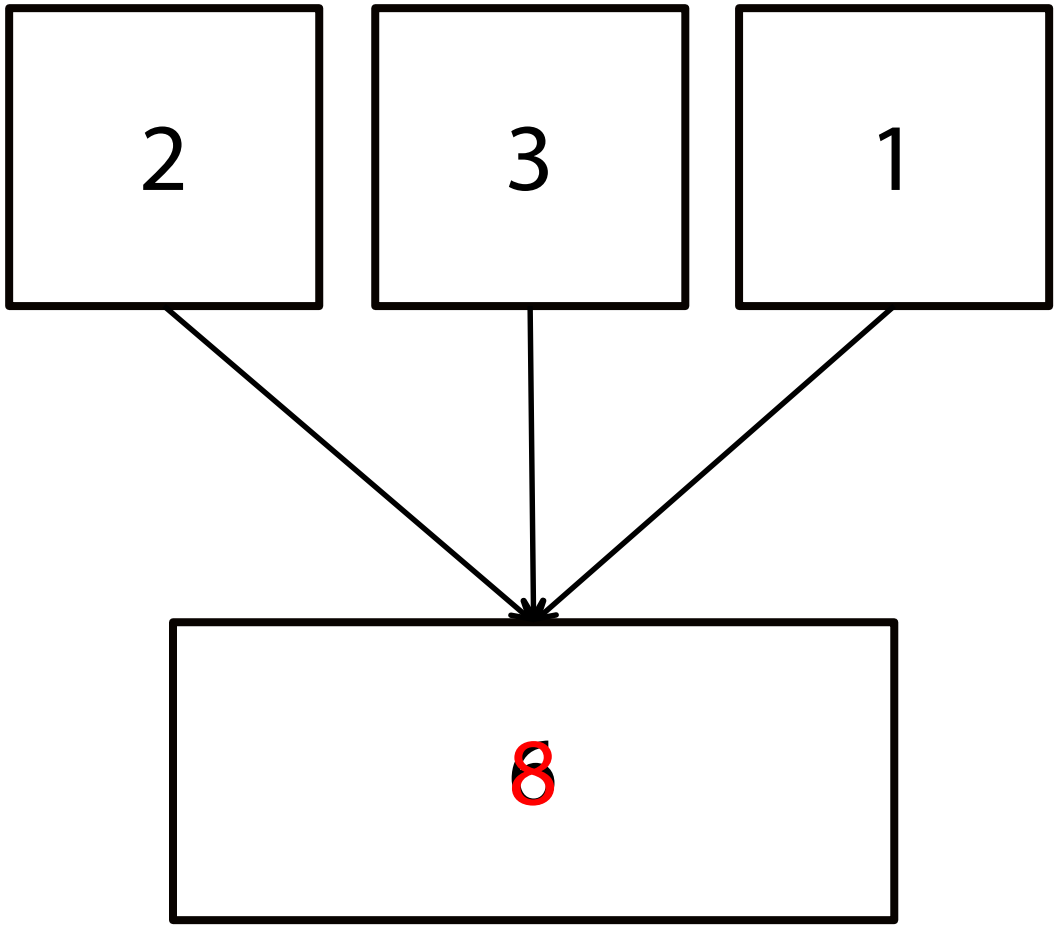
```
val accumulatedValue = accumulator.value
```

Driver

Accumulator



Accumulator



Accumulator

```
rdd.foreach(x=>{
  try{
    fn(x)
  }
  catch {
    case _ => errorCounterAccumulator += 1
  }
}
```

Java

```
JavaRDD.[INSERT METHOD OF CHOICE](
  new Function<TIn, TOut>(){
    public TOut call(TIn value){
      //process(value) -> TOut
    }
  }
)
```

Java

Function<TIn, TOut>	TOut call(TIn value)
Function2<TIn1, TIn2, TOut>	TOut call(TIn1 value1, TIn2 value2)
Function3<TIn1, TIn2, TIn3, TOut>	TOut call(TIn1 value1, TIn2 value2, TIn3 value3)
FlatMapFunction<TIn, TOut>	Iterable<TOut> call(TIn value)
FlatMapFunction2<TIn1, TIn2, TOut>	Iterable<TOut> call(TIn1 value1, TIn2 value2)
VoidFunction<TIn>	void call(TIn value)
PairFunction<TIn, TKey, TValue>	Tuple2<TKey, TValue> call(TIn value)
PairFlatMapFunction<TIn, TKey, TValue>	Iterable<Tuple2<TKey, TValue> call(TIn value)

org.apache.spark.api.java.function

Java

```
JavaPairRDD<TK, TV> mapToPair<TK, TV> (PairFunction<TIn, TK, TV>)
```

Resources

- Official Documentation
  - https://spark.apache.org/docs/latest/programming-guide.html
- Python API: Josh Rosen
  - https://www.youtube.com/watch?v=xc7Lc8RA8wE

Summary

- Caching
- Accumulators
- Java