LuceneRDD

Entity Linkage & (Spatial) Search

Anastasios Zouzias

ScalaIO @ Lyon 2016

About myself

* **Data Scientist @ Swisscom, CH:** Mobility Insights Team
* Researcher @ IBM Research, Zurich: Machine Learning & Search Engineer
* PhD in Univ. of Toronto: ML & Randomized Algorithms
* Scala Developer (day job): 2-3 years
* Apache Solr / Elasticsearch developer: 3+ years

How many of you have used **Spark**?

ebiznext

Zengularity.

Valraiso

Teads

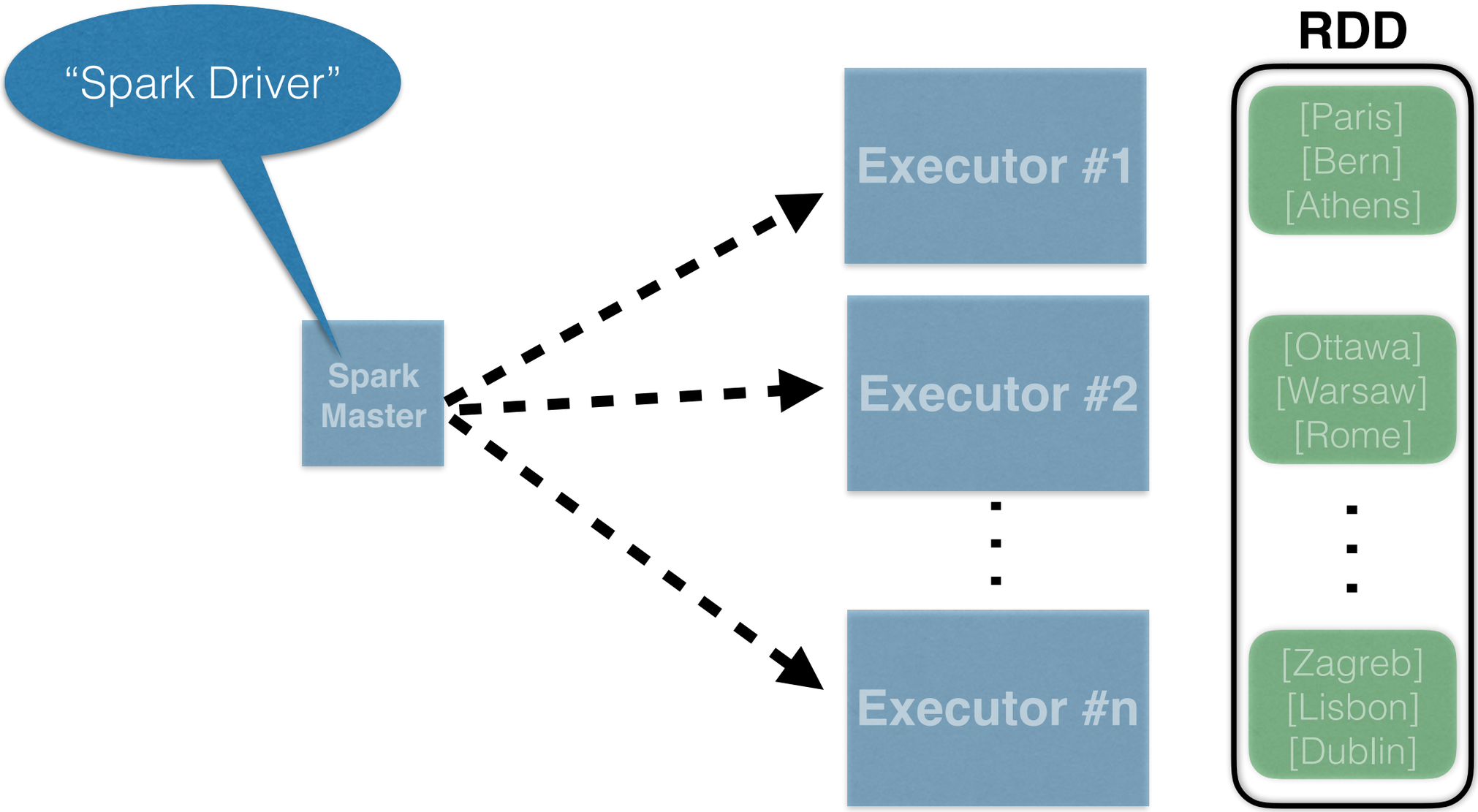
TABMO GROUP

mfg labs

LUNATECH

criteo labs

Apache Spark



RDD (Resilient Distributed Dataset)
Distributes data over multiple nodes

How many of you have used Lucene/Solr/Elastic?

Apache Lucene

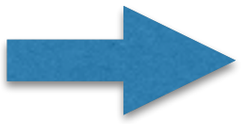
* **Lucene** is a powerful **Java search library** that lets you easily do **search** or **Information Retrieval (IR)**
* Used by LinkedIn, Twitter, and many more…
* Scalable and High-performance indexing
* Powerful, Accurate and Efficient Search algorithms

ebiznext Zengularity. Valraiso Teads TABMOGROUP mfg labs LUNATECH criteo labs

LuceneRDD

What is it?

Spark:
Partitioning / distribution of queries / data



LuceneRDD:
Query / dispatch / aggregation of data

Lucene:
Indexing / querying data (single partition)

Open-source project
https://github.com/zouzias/spark-lucenerdd

ebiznext

Zengularity. Valraiso Teads TABMO GROUP mfg labs LUNATECH criteo labs

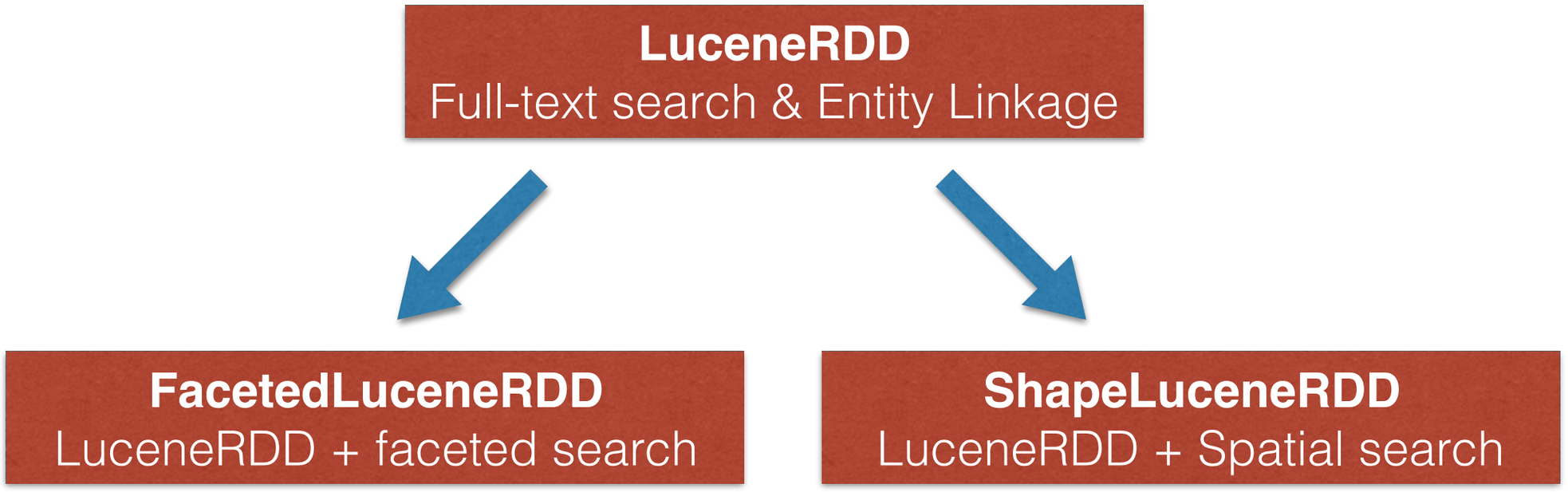
Motivation

Why **LuceneRDD** exists?

* **Natively** support of full-text / spatial search in Spark (without external Elastic/Solr cluster)
* **Scalable** Entity Linkage (approx. join) with Spark & Lucene
* Personal: Better understanding of Spark's Internals (RDDs)
* Personal: Better understanding of Scala (implicits)

ebiznext Zengularity. Valraiso Teads TABMO GROUP mfg labs LUNATECH criteo labs

LuceneRDD: RDD with search



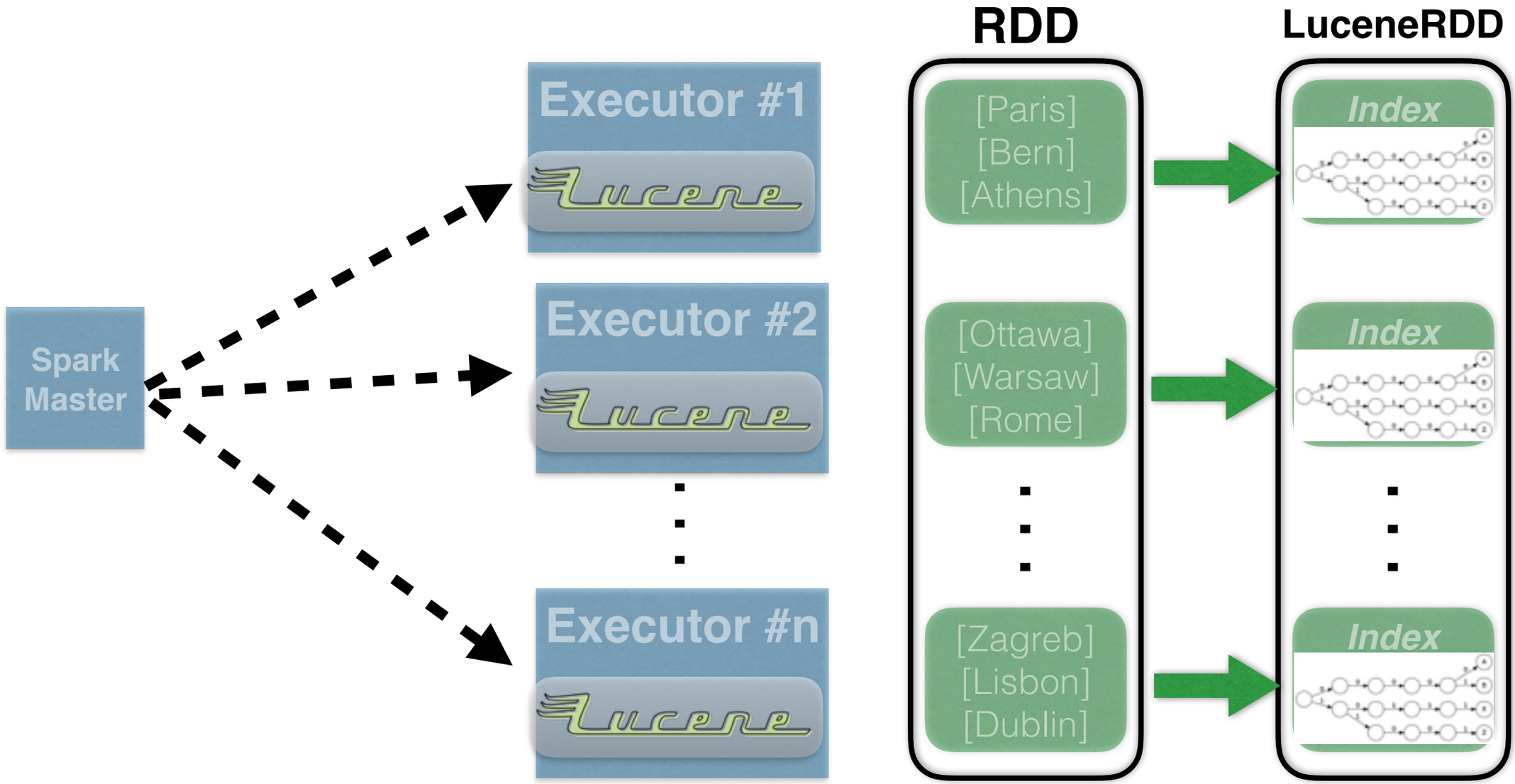

```
spark-shell --packages org.zouzias:spark-lucenerdd:0.2.2
```

or

```
libraryDependencies +=  "org.zouzias" %% "spark-lucenerdd" % "0.2.2"
```

* Main development: Spark **2.x** (supports Spark >= **1.4**)
* Lucene **5.5.3** (Lucene **6.2.2** JVM 8)
* Released on maven central & spark-packages (Scala 2.10 / 2.11)

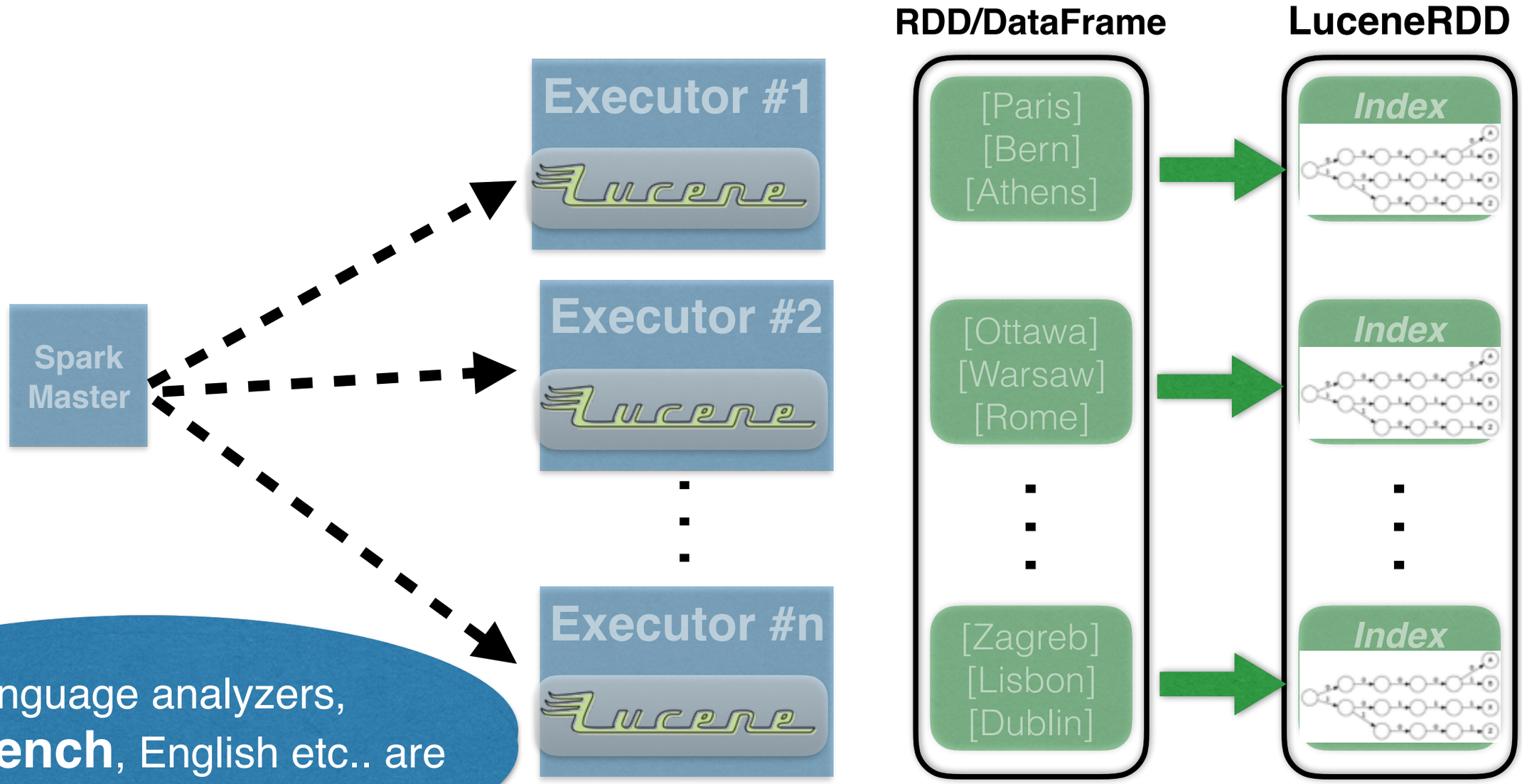
LuceneRDD



Main idea
Index RDD partition data with distributed Lucene index (**inverted index per partition**)

Index storage **disk** or **memory**

DataFrame to LuceneRDD*

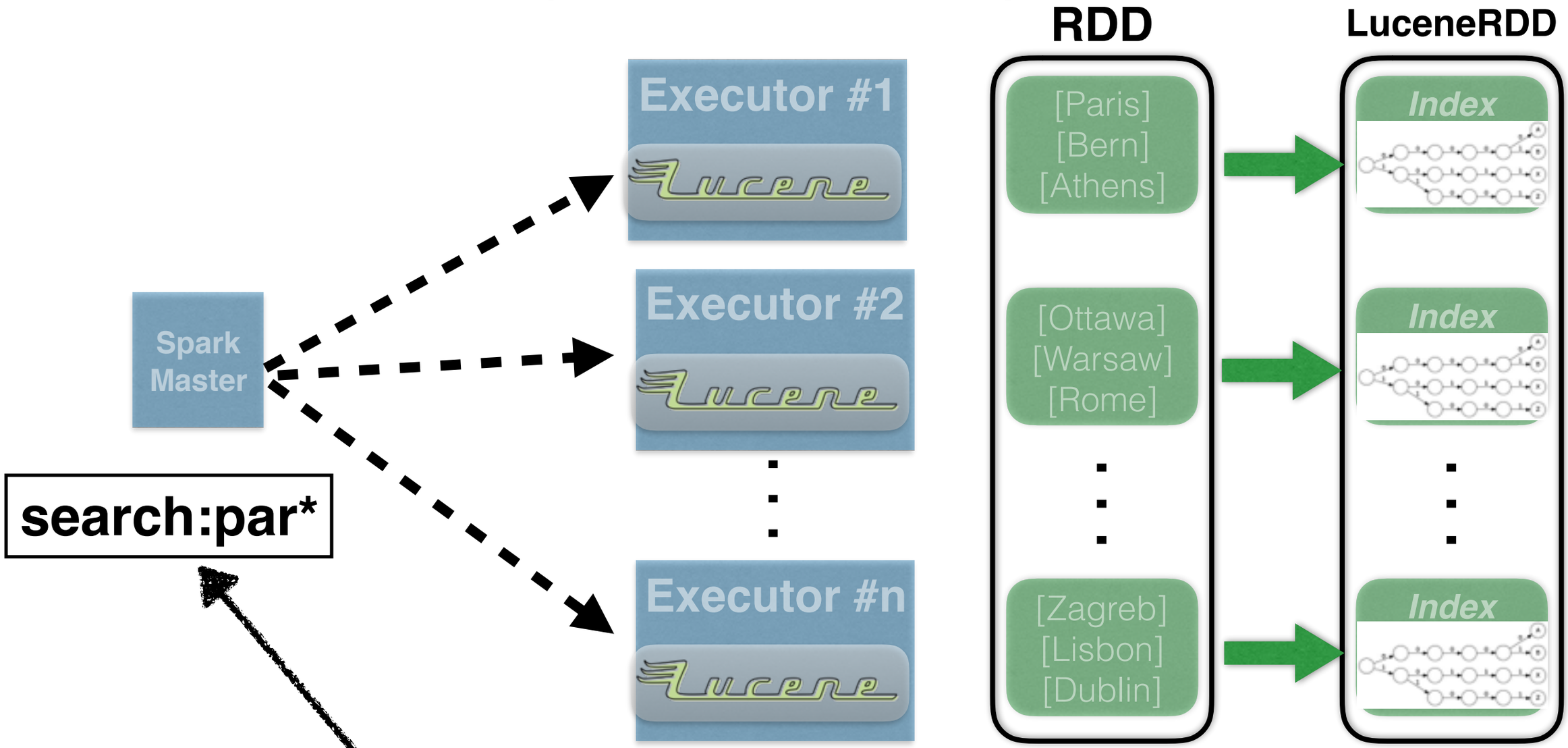


Language analyzers, i.e., **French**, English etc.. are configurable

Load all capitals

```
val capitals = spark.read.parquet("data/spatial/capitals.parquet").select("name", "country")
val luceneRDD = LuceneRDD(capitals)
```

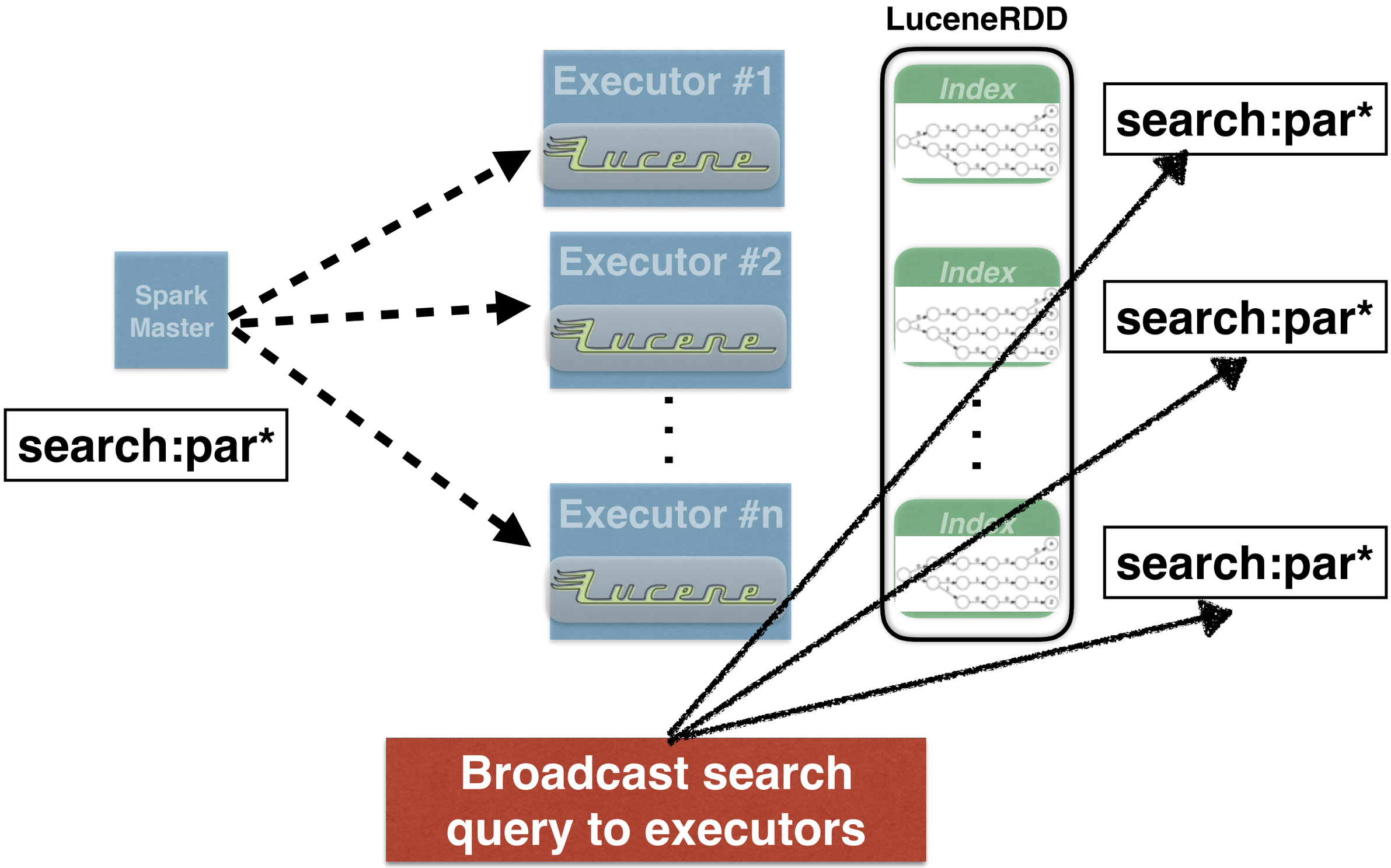
Simplistic Example



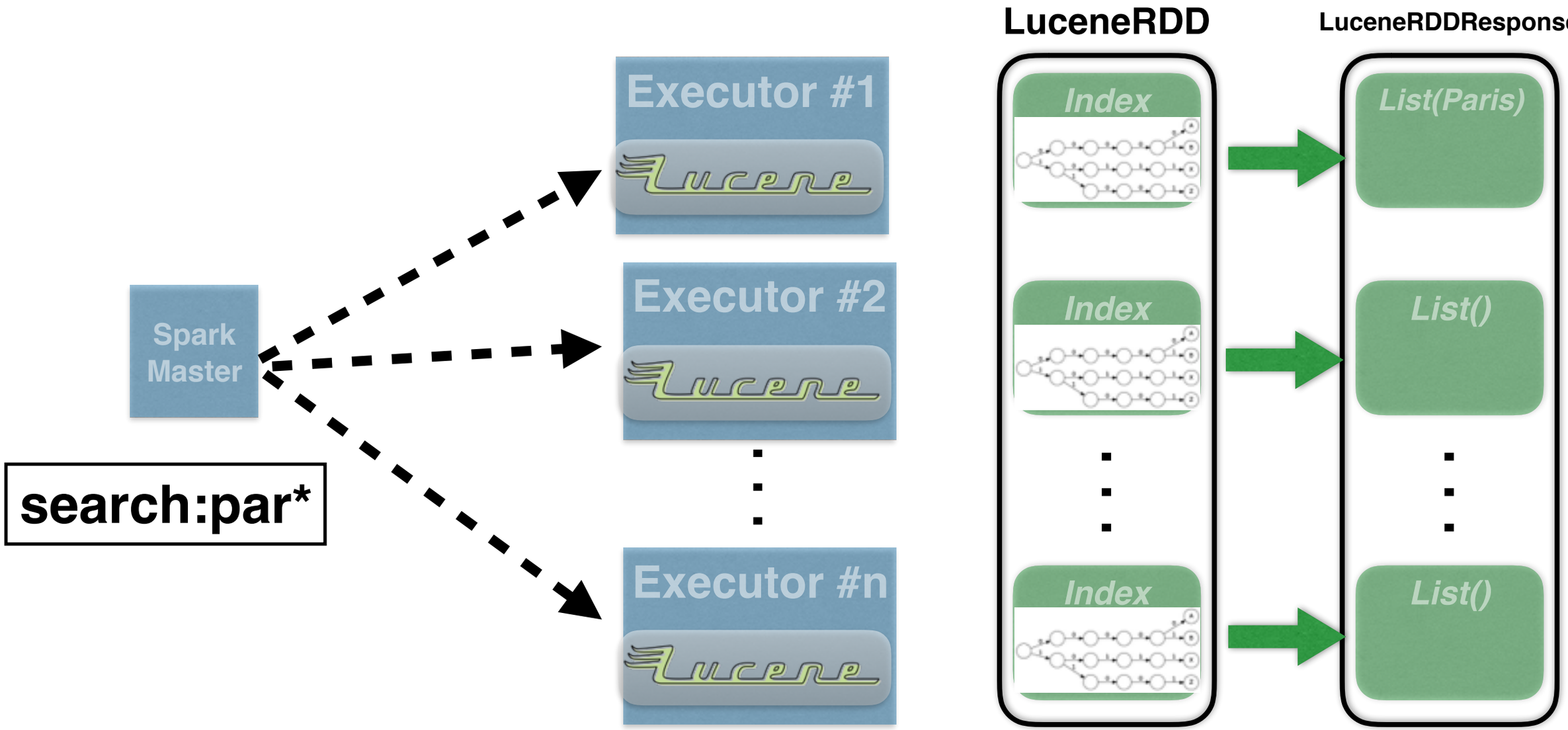
Prefix query for "par"

```
val topK = 10
val response = luceneRDD.prefixQuery("name", "par", topK)
```

Simplistic Example



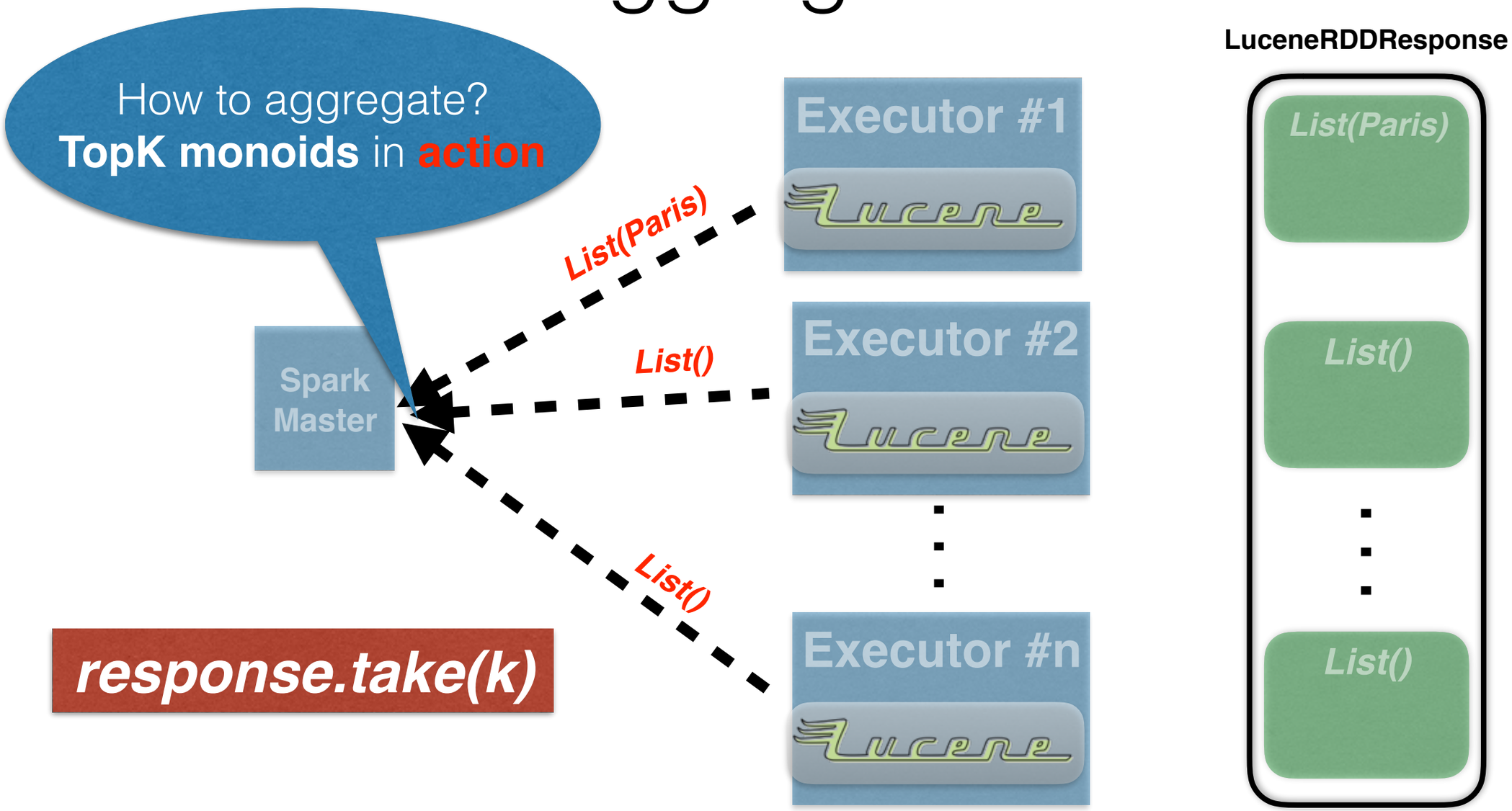
Simplistic Example



topK results / partition

Lucene prefix Search on each partition

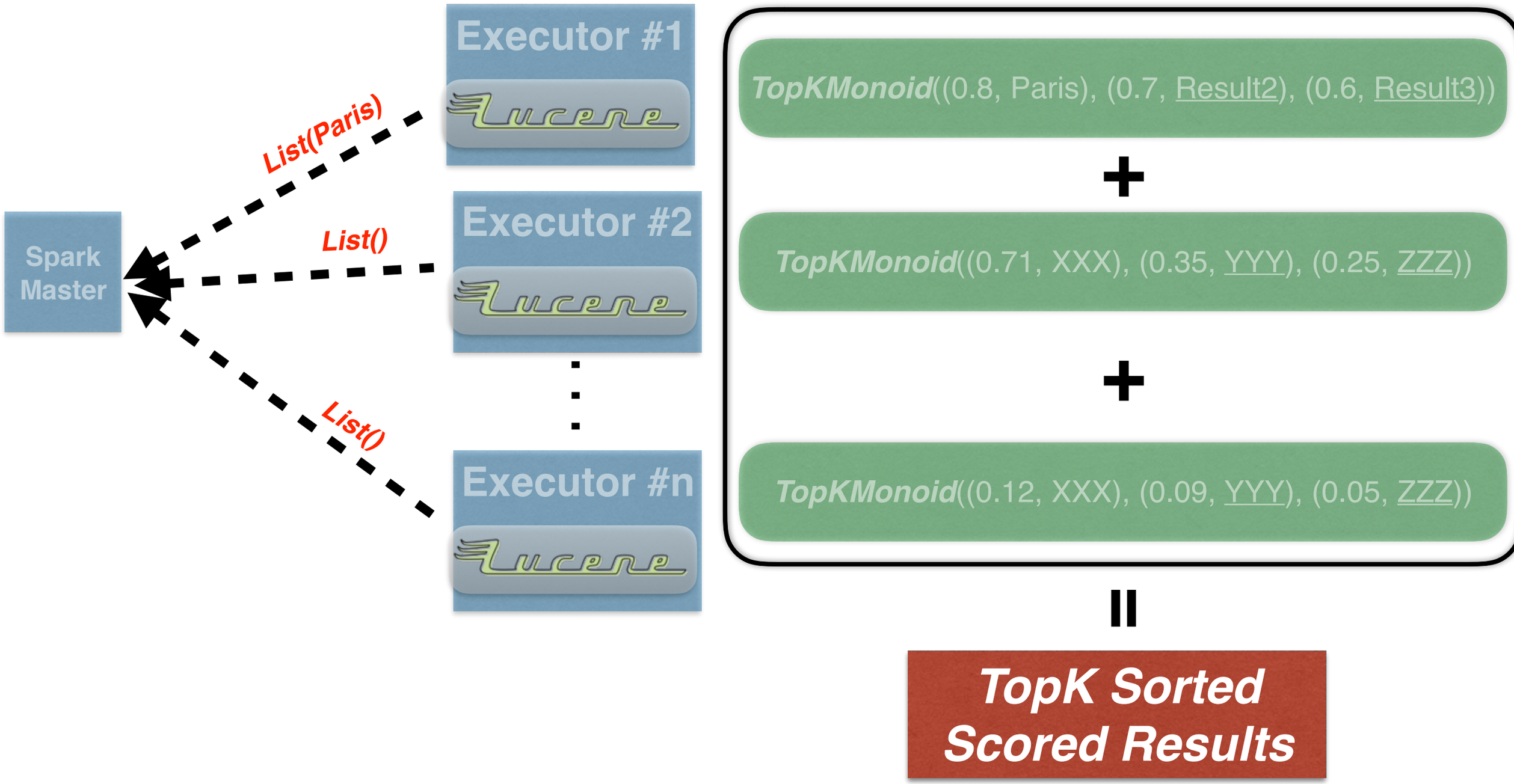
Aggregate Results



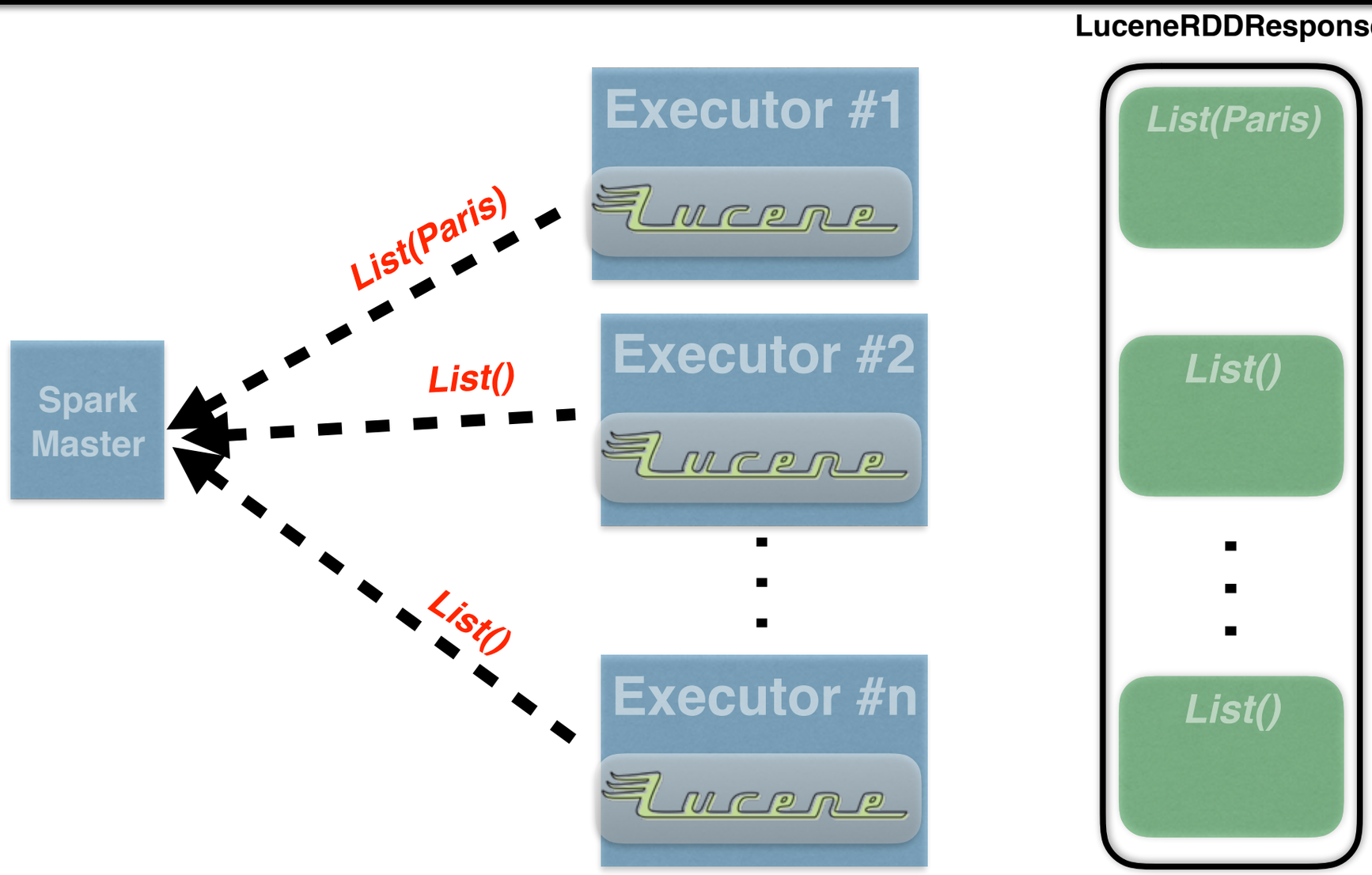
response.take(k)

```
response.take(k).foreach(println)

SparkScoreDoc(1.0,9,0,Text fields:country:[France]name:[Paris])
SparkScoreDoc(1.0,43,0,Text fields:country:[Suriname]name:[Paramaribo])
```

Aggregate Results



Twitter's **Algebird** TopKMonoid



LuceneRDDResponse.take(k)

```scala
/**
 * Use [[TopKMonoid]] to take
 * @param num
 * @return
 */
override def take(num: Int): Array[SparkScoreDoc] = {
  val monoid = new TopKMonoid[SparkScoreDoc](num)(ordering)
  partitionsRDD.map(monoid.build(_))
    .reduce(monoid.plus).items.toArray
}
```

LuceneRDD Operations

Operation	Syntax	Description
Term Query	`LuceneRDD.termQuery(field, query, topK)`	Exact term search
Fuzzy Query	`LuceneRDD.fuzzyQuery(field, query, maxEdits, topK)`	Fuzzy term search
Phrase Query	`LuceneRDD.phraseQuery(field, query, topK)`	Phrase search
Prefix Query	`LuceneRDD.prefixSearch(field, prefix, topK)`	Prefix search
Query Parser	`LuceneRDD.query(queryString, topK)`	Query parser search
Faceted Search	`FacetedLuceneRDD.facetQuery(queryString, field, topK)`	Faceted Search
Record Linkage	`LuceneRDD.link(otherEntity: RDD[T], linkageFct: T => searchQuery, topK)`	Record linkage via Lucene queries
Circle Search	`ShapeLuceneRDD.circleSearch((x,y), radius, topK)`	Search within radius
Bbox Search	`ShapeLuceneRDD.bboxSearch(lowerLeft, upperLeft, topK)`	Bounding box
Spatial Linkage	`ShapeLuceneRDD.linkByRadius(RDD[T], linkage: T => (x,y), radius, topK)`	Spatial radius linkage

Flexible multi-fields query:
term/Fuzzy/Prefix/etc sub-queries
+ boolean queries

FacetedLuceneRDD
LuceneRDD + faceted search

ShapeLuceneRDD
LuceneRDD + Spatial search

ebiznext

Zengularity. Valraiso Teads TABMOGROUP mfg labs LUNATECH criteo labs

Entity Linkage

a.k.a. approximate join

Entity Linkage (approx. join)

Left Dataset

James Stephenson One Ford Way, Dearborn, MI
Ford Petaprod

Alicia Thomson 4789 Woodward Avenue Detroit
Detroitics MegaProd,TeraProd

Jack Jones One Microsoft Way Redmond
Microsoft USA PetaProd

James Jones 1234 Woodward Ave. Detroit *Home*, MI
Moonlighting Mega-Prod

James Jones 4789 Woodward Ave. Detroit *Work*, MI
Detroitics GigaProd,MegaProd

Al Shepard New York, NY
Newyorkonics SuperProd,MegaProd

Mary Barry GM Renaissance Center, Detroit, MI 48243
General Motors TeraProd,SuperProd

Ram Kumar New Orchard Road, Armonk, NY
International Business Machines PetaProduct

Joel Smith New York, NY
New York-onics MegaProduct,SuperProd

Mike Taylor Unknown
Lockheed-M Tera Product

?

Right Dataset

James (Jim) Stephenson One Ford Way, Dearborn, MI 48126
Ford Petaprod

Alice Thompson 4789 Woodward Ave, Detroit, MI
Detroitics MegaProduct,TeraProd

Jackob Jones One Microsoft Way Redmond, WA 98052-7329
Microsoft Peta Product

Jim Jones 1234 Woodward Ave. Detroit, MI
Moonlighting, Inc MegaProd

James Jones 4789 Woodward Ave. Detroit, MI
Detroitics GigaProd

Joe Smith 1234 56 St. New York, NY
Newyorkonics SuperProd,MegaProd

Mary Barry Renaissance Center, Detroit
GM Tera product

Ramkrishna (Ram) Kumar New Orchard Road, Armonk, New York 10504
IBM GigaProduct, PetaProduct

Alicia Shepard 1234 56 St. New York, NY
Newyorkonics ???

Michael Taylor Classified, Bethesda, MD
Lockheed Martin TeraProd

Join left & **right** dataset
according to a “similarity”
(NO ids available)

Simplicity:
one-to-one linkage

Entity Linkage (approx. join)

Left Dataset

James Stephenson One Ford Way, Dearborn, MI
Ford Petaprod

Alicia Thomson 4789 Woodward Avenue Detroit
Detroitics MegaProd,TeraProd

Jack Jones One Microsoft Way Redmond
Microsoft USA PetaProd

James Jones 1234 Woodward Ave. Detroit *Home*, MI
Moonlighting Mega-Prod

James Jones 4789 Woodward Ave. Detroit *Work*, MI
Detroitics GigaProd,MegaProd

Al Shepard New York, NY
Newyorkonics SuperProd,MegaProd

Mary Barry GM Renaissance Center, Detroit, MI 48243
General Motors TeraProd,SuperProd

Ram Kumar New Orchard Road, Armonk, NY
International Business Machines PetaProduct

Joel Smith New York, NY
New York-onics MegaProduct,SuperProd

Mike Taylor Unknown
Lockheed-M Tera Product

?

Right Dataset

James (Jim) Stephenson One Ford Way, Dearborn, MI 48126
Ford Petaprod

Alice Thompson 4789 Woodward Ave, Detroit, MI
Detroitics MegaProduct,TeraProd

Jackob Jones One Microsoft Way Redmond, WA 98052-7329
Microsoft Peta Product

Jim Jones 1234 Woodward Ave. Detroit, MI
Moonlighting, Inc MegaProd

James Jones 4789 Woodward Ave. Detroit, MI
Detroitics GigaProd

Joe Smith 1234 56 St. New York, NY
Newyorkonics SuperProd,MegaProd

Mary Barry Renaissance Center, Detroit
GM Tera product

Ramkrishna (Ram) Kumar New Orchard Road, Armonk, New York 10504
IBM GigaProduct, PetaProduct

Alicia Shepard 1234 56 St. New York, NY
Newyorkonics ???

Michael Taylor Classified, Bethesda, MD
Lockheed Martin TeraProd

Naive approach
Compute pairwise scores does not scale: **O(n²)**

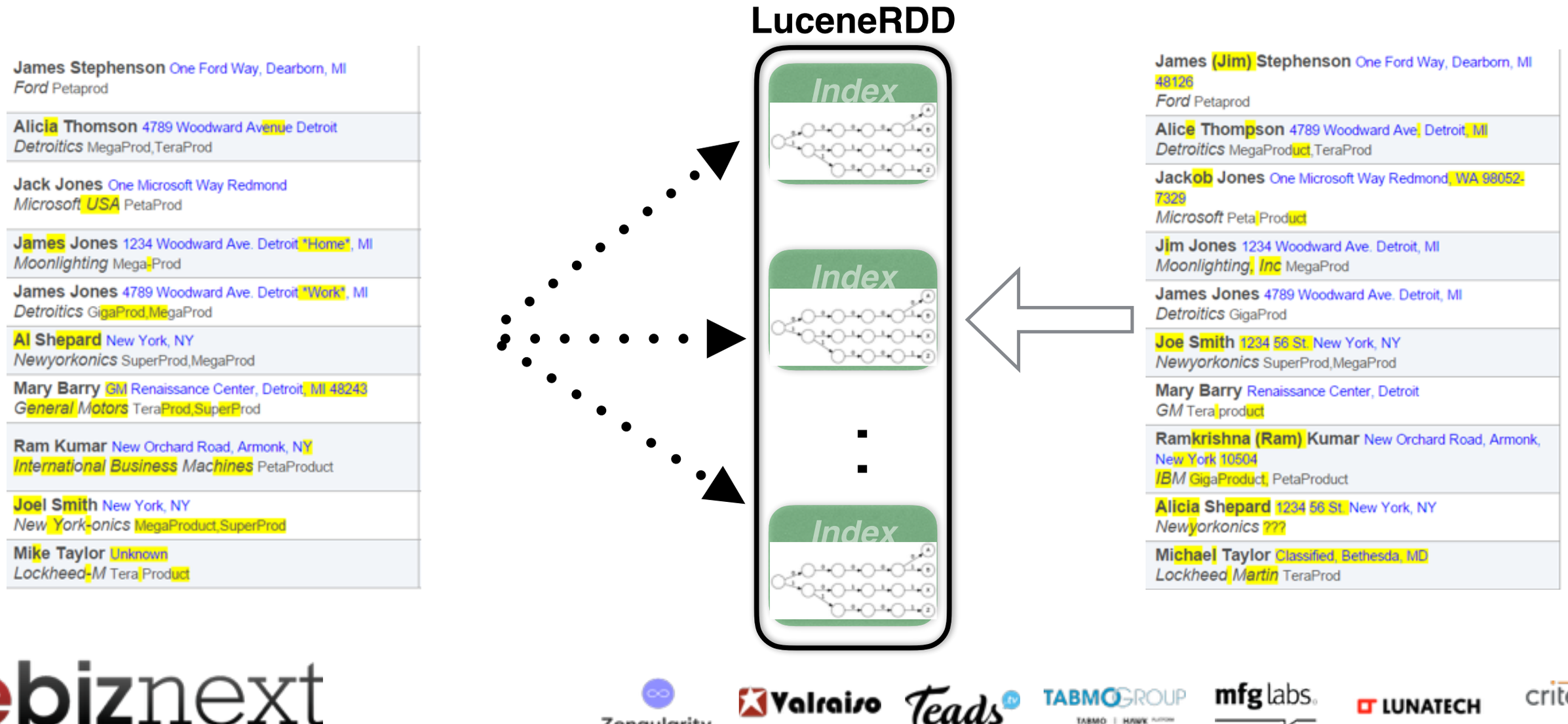
Entity Linkage API

Operation	Syntax	Description
Term Query	`LuceneRDD.termQuery(field, query, topK)`	Exact term search
Fuzzy Query	`LuceneRDD.fuzzyQuery(field, query, maxEdits, topK)`	Fuzzy term search
Phrase Query	`LuceneRDD.phraseQuery(field, query, topK)`	Phrase search
Prefix Query	`LuceneRDD.prefixSearch(field, prefix, topK)`	Prefix search
Query Parser	`LuceneRDD.query(queryString, topK)`	Query parser search
Faceted Search	`FacetedLuceneRDD.facetQuery(queryString, field, topK)`	Faceted Search
Record Linkage	`LuceneRDD.link(otherEntity: RDD[T], linkageFct: T => searchQuery, topK)`	Record linkage via Lucene queries
Circle Search	`ShapeLuceneRDD.circleSearch((x,y), radius, topK)`	Search within radius
Bbox Search	`ShapeLuceneRDD.bboxSearch(lowerLeft, upperLeft, topK)`	Bounding box
Spatial Linkage	`ShapeLuceneRDD.linkByRadius(RDD[T], linkage: T => (x,y), radius, topK)`	Spatial radius linkage

LuceneRDD Approximate Join

1) Index **right** dataset using **LuceneRDD**
2) Define "search similarity function" between **each left entity** & **right** dataset
3) For every **left** entity, **search** & **zip** using **topK** results

Record Linkage	`LuceneRDD.link(otherEntity: RDD[T], linkageFct: T => searchQuery, topK)`	Record linkage via Lucene queries



James Stephenson One Ford Way, Dearborn, MI
Ford Petaprod

Alicia Thomson 4789 Woodward Avenue Detroit
Detroitics MegaProd,TeraProd

Jack Jones One Microsoft Way Redmond
Microsoft USA PetaProd

James Jones 1234 Woodward Ave. Detroit *Home*, MI
Moonlighting MegaProd

James Jones 4789 Woodward Ave. Detroit *Work*, MI
Detroitics GigaProd,MegaProd

Al Shepard New York, NY
Newyorkonics SuperProd,MegaProd

Mary Barry GM Renaissance Center, Detroit, MI 48243
General Motors TeraProd,SuperProd

Ram Kumar New Orchard Road, Armonk, NY
International Business Machines PetaProduct

Joel Smith New York, NY
New York-onics MegaProduct,SuperProd

Mike Taylor Unknown
Lockheed-M TeraProduct

(3a) Map to search queries

RDD[String]

List("name:james")

List("name:joel")

List("name:ram")

(3b) Collect & broadcast (**RDD[String]**)

Spark Master

LuceneRDD

Index

Index

Index

James (Jim) Stephenson One Ford Way, Dearborn, MI 48126
Ford Petaprod

Alice Thompson 4789 Woodward Ave Detroit, MI
Detroitics MegaProduct,TeraProd

Jackob Jones One Microsoft Way Redmond, WA 98052-7329
Microsoft PetaProduct

Jim Jones 1234 Woodward Ave. Detroit, MI
Moonlighting, Inc MegaProd

James Jones 4789 Woodward Ave. Detroit, MI
Detroitics GigaProd

Joe Smith 1234 56 St. New York, NY
Newyorkonics SuperProd,MegaProd

Mary Barry Renaissance Center, Detroit
GM TeraProduct

Ramkrishna (Ram) Kumar New Orchard Road, Armonk, New York 10504
IBM GigaProduct, PetaProduct

Alicia Shepard 1234 56 St. New York, NY
Newyorkonics ???

Michael Taylor Classified, Bethesda, MD
Lockheed Martin TeraProd

LuceneRDD Entity Linkage

1) ~~Index **right** dataset using **LuceneRDD**~~
2) ~~Define "search similarity function" between **left** & **right** dataset~~
3) For every left entity, **search** & **zip** by search similarly

ebiznext Zengularity. Valraiso Teads.tv TABMOGROUP mfg labs LUNATECH criteo labs

neRDD: linkage

Query each **partition** and **zip** query with **topK** results

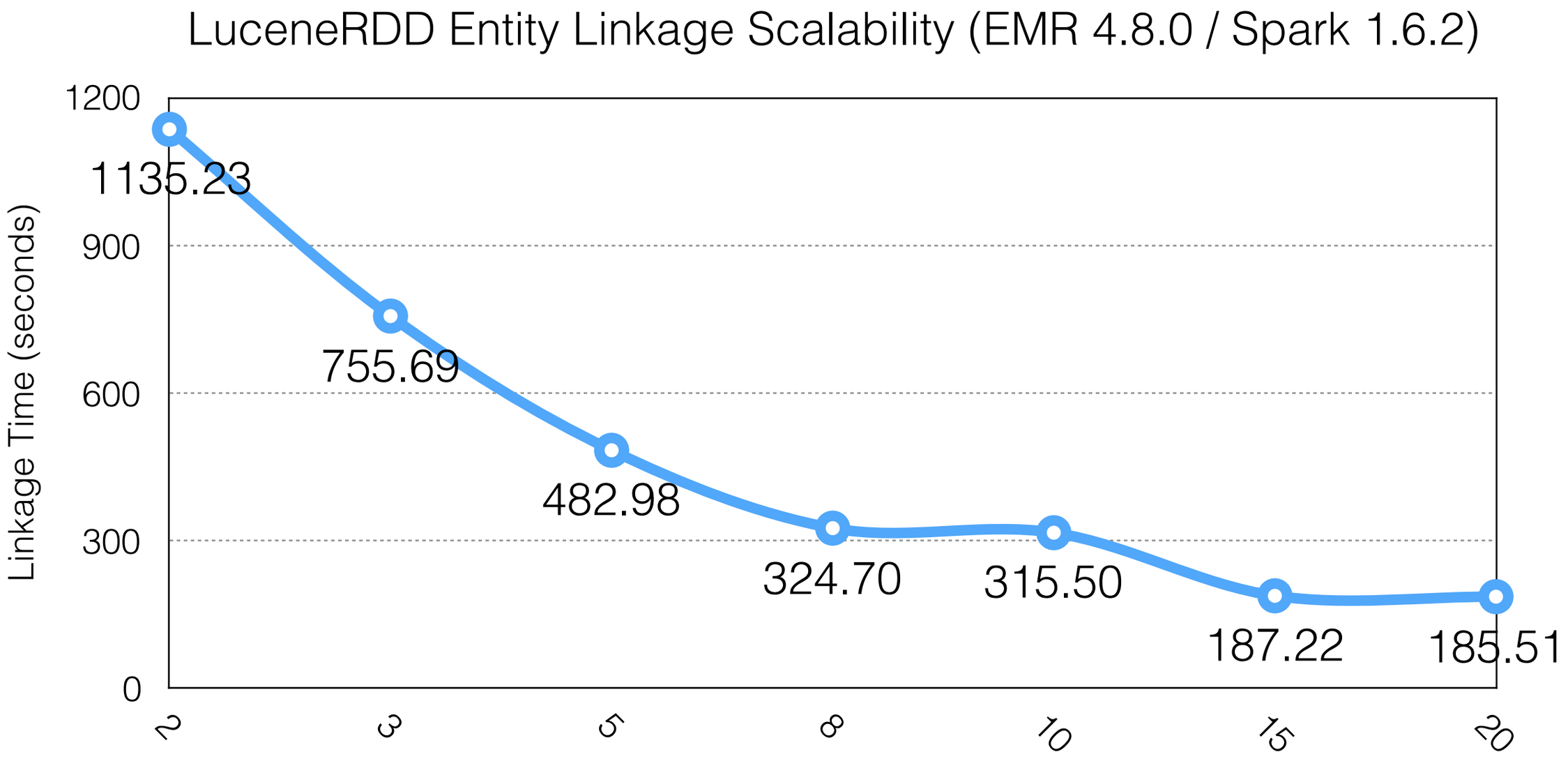
Map each **left** entity to search query & **broadcast** to each partition

```
def link[ : ClassTag](other: RDD[T1], searchQueryGen: T1 => String pK: Int = DefaultTopK)
  : RDD[ , List[SparkScoreDoc])] = {
  val mo oid = new TopKMonoid[SparkScoreDoc](topK)(SparkScoreD .descending)

  val  eries = other.map(searchQueryGen).collect()
  val ueriesB = partitionsRDD.context.broadcast(queries)

  val resultsByPart: RDD[(Long, TopK[SparkScoreDoc])] = partitionsRDD.flatMap { partition =>
    queriesB.value.zipWithIndex.map { case (qr, index) =>
      val results = partition.query(qr, topK).map(x => monoid.build(x))
      (index.toLong, results.reduceOption(monoid.plus).getOrElse(monoid.zero))
    }
  }

  val results = resultsByPart.reduceByKey(monoid.plus)
  other.zipWithIndex.map(_.swap).join(results).values
    .map(joined => (joined._1, joined._2.items.take(topK)))
}
```

Reduce **topK** monoids per query

Does LuceneRDD approx. join scale?

https://github.com/zouzias/spark-lucenerdd-aws



Joined Datasets

* H1B US Visa Applications (2.6 million records)
* Geonames US Cities (4.4 million records)

AWS: master: m3.xlarge, Core: r3.xlarge, Exec. mem: 9G

Demo

LuceneRDD Linkage API (ACM vs DBLP research articles)

Köpcke, H.; Thor, A.; Rahm, E.
Evaluation of entity resolution approaches on real-world match problems
Proc. 36th Intl. Conference on Very Large Databases (VLDB), 2010

https://github.com/zouzias/spark-lucenerdd-examples

Spatial/Entity Linkage API

Operation	Syntax	Description
Term Query	`LuceneRDD.termQuery(field, query, topK)`	Exact term search
Fuzzy Query	`LuceneRDD.fuzzyQuery(field, query, maxEdits, topK)`	Fuzzy term search
Phrase Query	`LuceneRDD.phraseQuery(field, query, topK)`	Phrase search
Prefix Query	`LuceneRDD.prefixSearch(field, prefix, topK)`	Prefix search
Query Parser	`LuceneRDD.query(queryString, topK)`	Query parser search
Faceted Search	`FacetedLuceneRDD.facetQuery(queryString, field, topK)`	Faceted Search
Record Linkage	`LuceneRDD.link(otherEntity: RDD[T], linkageFct: T => searchQuery, topK)`	Record linkage via Lucene queries
Circle Search	`ShapeLuceneRDD.circleSearch((x,y), radius, topK)`	Search within radius
Bbox Search	`ShapeLuceneRDD.bboxSearch(lowerLeft, upperLeft, topK)`	Bounding box
Spatial Linkage	`ShapeLuceneRDD.linkByRadius(RDD[T], linkage: T => (x,y), radius, topK)`	Spatial radius linkage

Demo

ShapeLuceneRDD Linkage
(Country polygons vs Capital points)

https://github.com/zouzias/spark-lucenerdd-examples

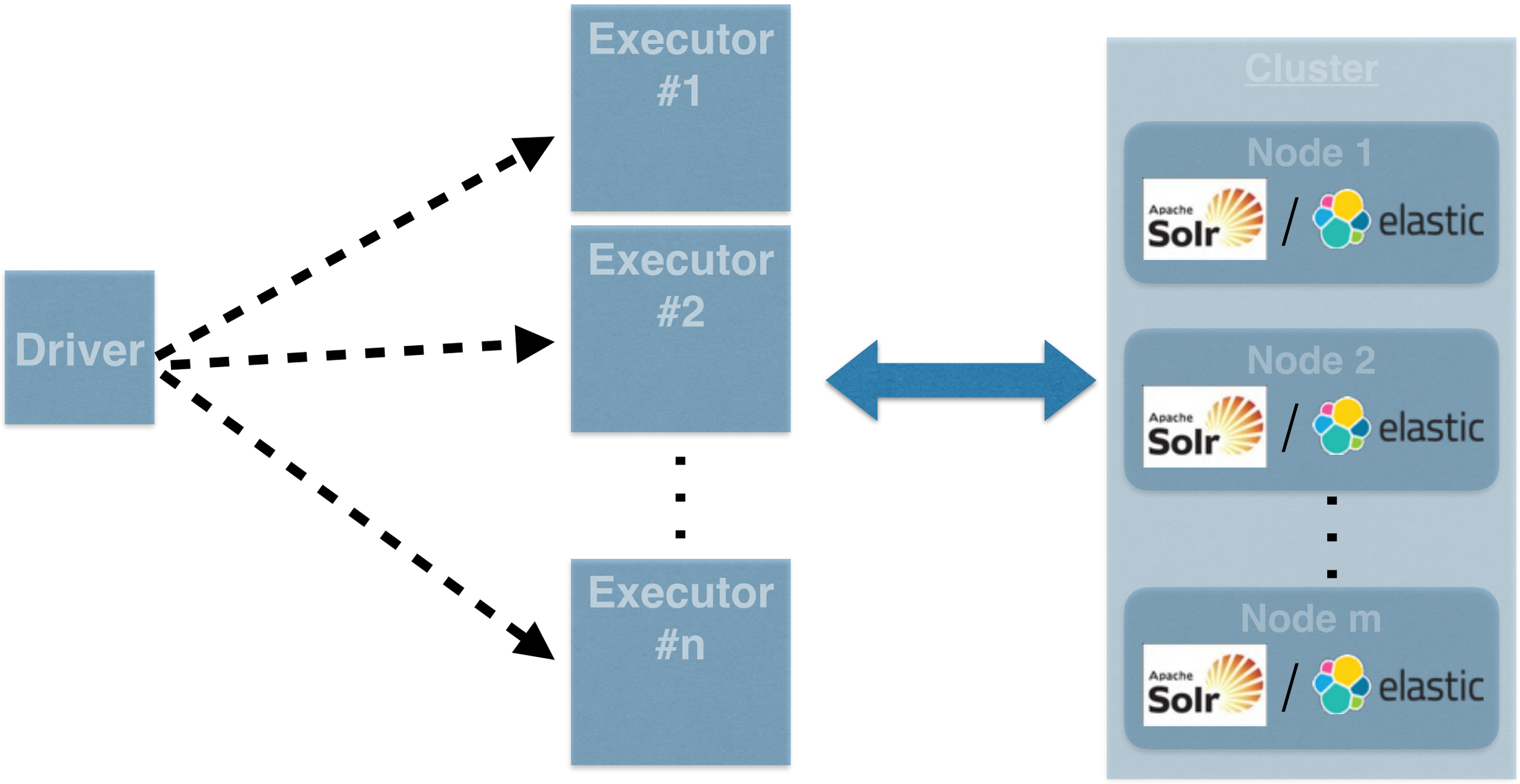
Future Work

* Contributors / use cases are welcome!
* More performance tests with approx. join
* Spatial-linkage performance tests using open Street Map data
* Bypass the collect-broadcast Spark pattern?

Merci pour votre attention!

Questions?

Elastic/SolrCloud Connector



Limitations

- **m** is **fixed** & (usually) small compared to **n**
- additional network communication: execs & elastic/solr