Lucidworks

gh-level Architecture

k

Solr

park SQL

g

committer since April 2014, work for

Cloud features … and bin/solr!

for Lucene / Solr 5.1

ction

nce working with Hadoop, Pig, Hive,
started using Spark about 6 months

rk.apache.org/

ww.cs.berkeley.edu/~matei/papers/2012/nsdi_spark.

ed alternative to MapReduce

sorted 100TB in 23 minutes (3x faster than Yahoo's previ

g Data

hms (PageRank, K-Means, Logistic regression) & inter

Spark Streaming

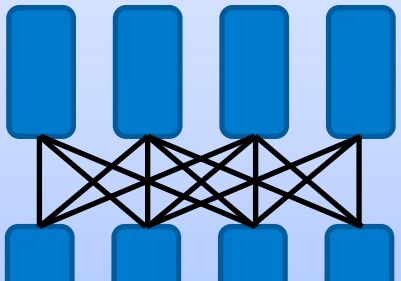
MLlib (machine learning)

Gr (E

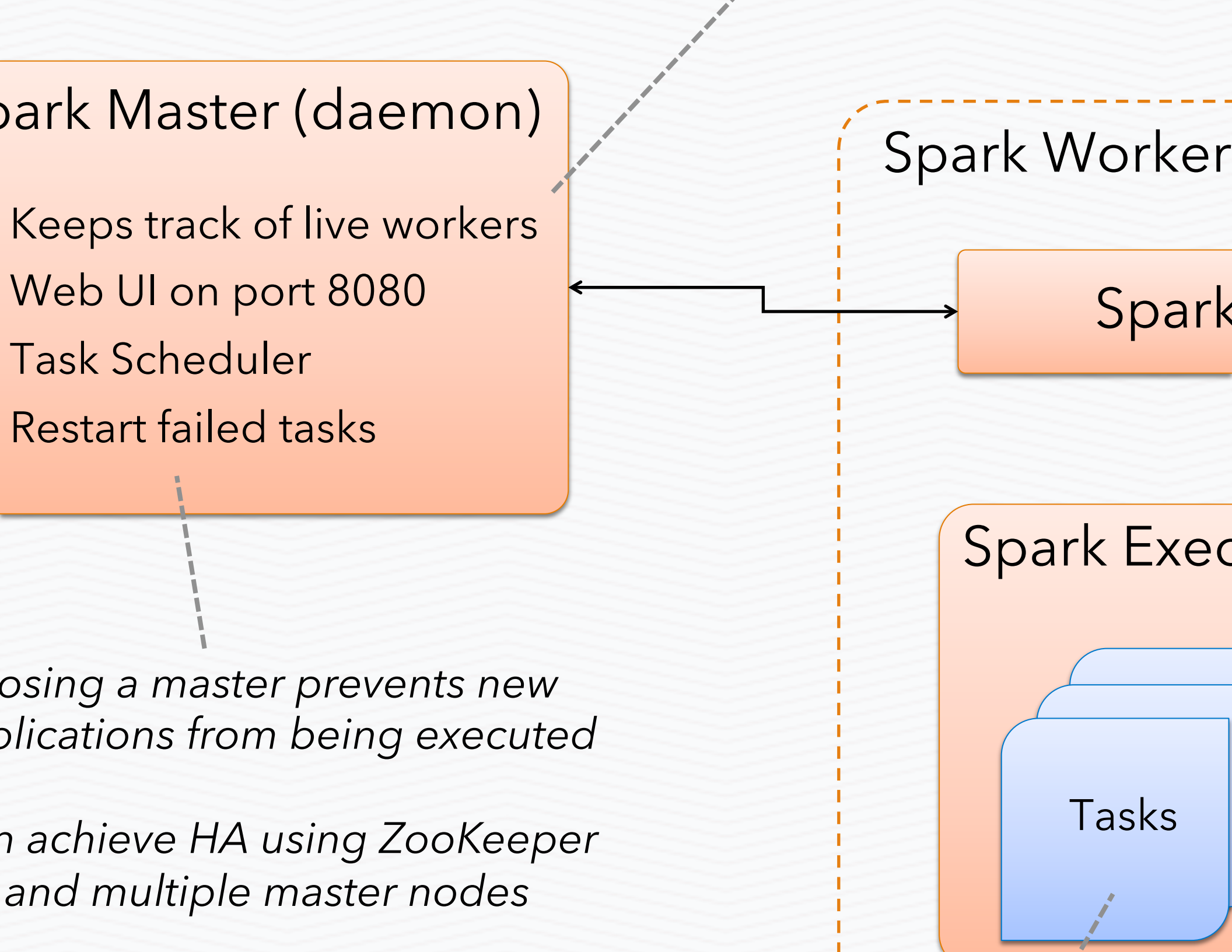
Spark Core

n

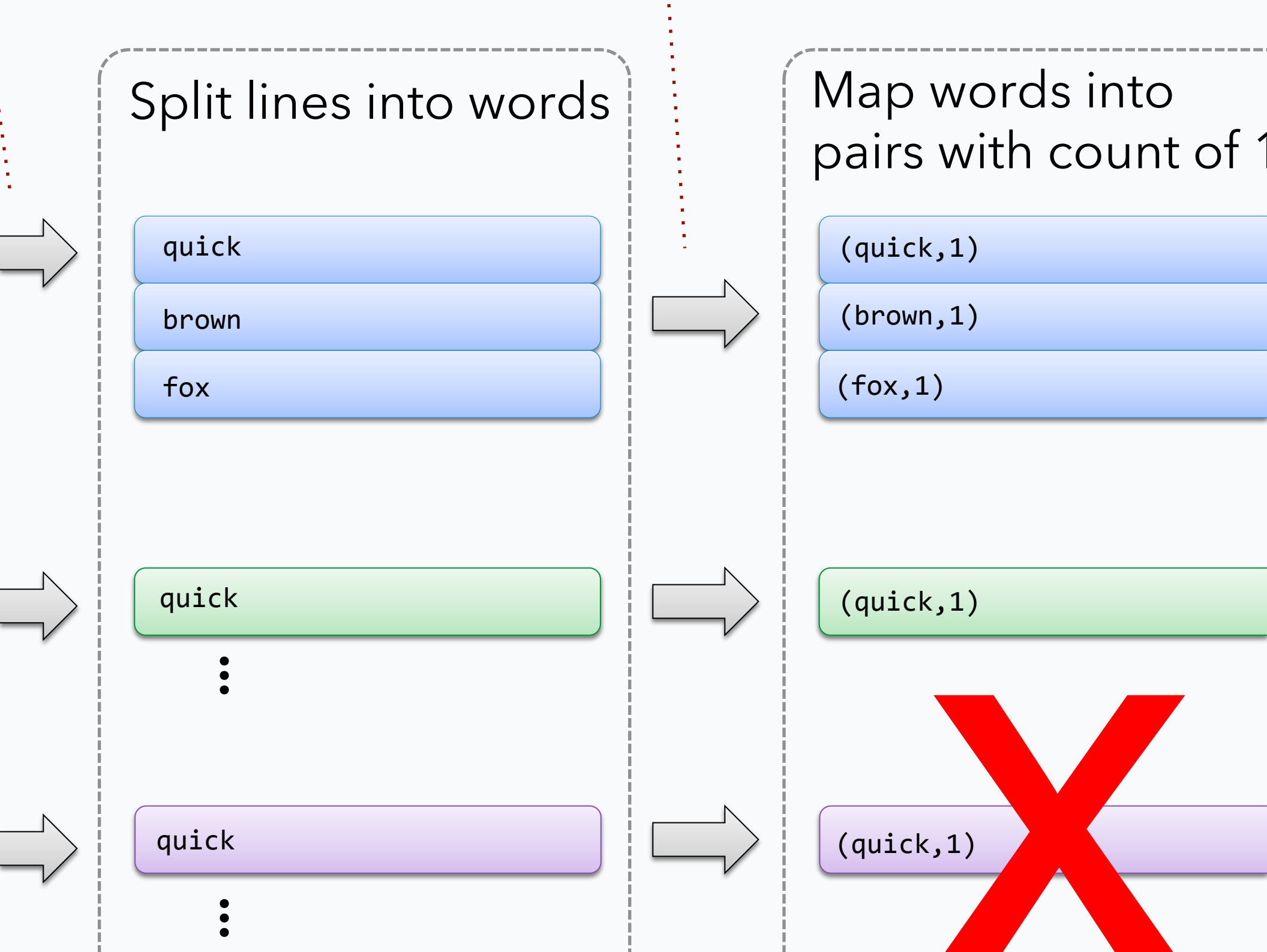
The Shuffle



Cachi


park Master (daemon)
Keeps track of live workers
Web UI on port 8080
Task Scheduler
Restart failed tasks
Spark Worker
Spark
Spark Exec
Tasks
osing a master prevents new
plications from being executed
n achieve HA using ZooKeeper
and multiple master nodes


```
val file = spark.textFile(
val counts = file.flatMap(
```


Split lines into words
Map words into pairs with count of 1
quick
brown
fox
quick
quick
(quick,1)
(brown,1)
(fox,1)
(quick,1)
(quick,1)

system OR using a transformation of an

e of coarse-grained transformations (ma

Ds can be re-computed by re-playing th

ersist an RDD (for reusing during interact

ons

ETL jobs

job

matching against stored queries

tions (interactive data mining, machi

lr query as Spark RDD (resilient distributed datas

ults from each shard in parallel

uence of RDDs

you can use all the other Spark libs (MLlib, etc)

es

must be less than the batch interval time

:

by, join, etc)

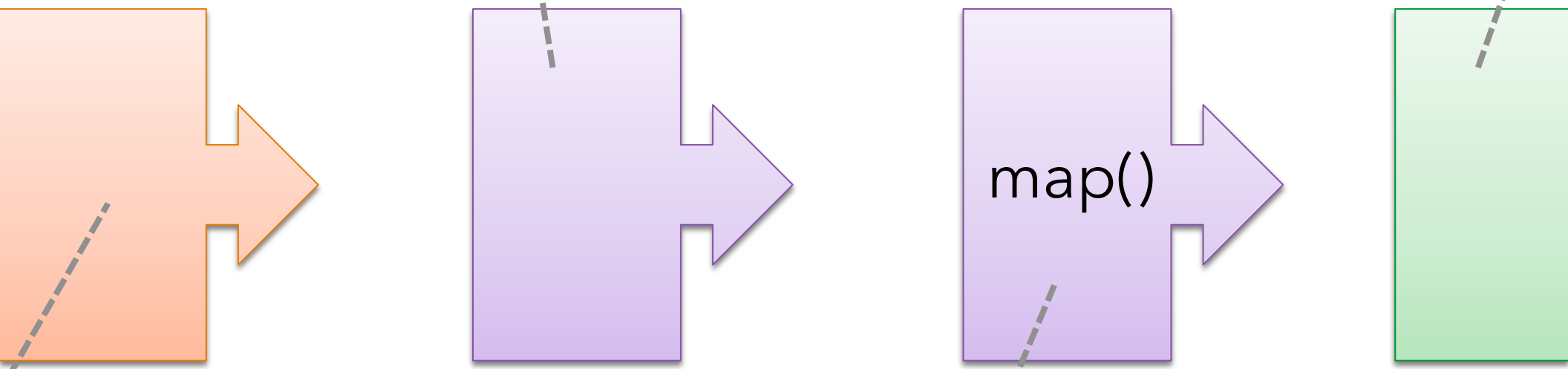
xternal sink, e.g. Solr)

e!

```
st localhost:2181 -collection social
```

Various transformations / enrichments on each tweet (e.g. sentiment analysis, language detection)

SolrSupport.index




```
us> tweets =
null, filters);
```

JavaDStream<SolrInputDocument> **docs** = **tweets**

```
tatus> tweets =
m(jssc, null, filters);

o SolrInputDocument objects for indexing in So
ent> docs = tweets.map(

InputDocument>() {
nt call(Status status) {
 =
oToSolrInputDoc("tweet-"+status.getId(), statu
er_s", "twitter");
_s", status.getUser().getScreenName());
", status.isRetweet() ? "echo" : "post");
```

```
utDocument>, Void>() {
olrInputDocument> solrInputDocumentJavaRDD) throws E
.foreachPartition(
tor<SolrInputDocument>>() {
rator<SolrInputDocument> solrInputDocumentIterator)
olrServer = getSolrServer(zkHost);
ment> batch = new ArrayList<SolrInputDocument>();
cumentIterator.hasNext()) {
putDocumentIterator.next());
 >= batchSize)
r(solrServer, collection, batch);


())
solrServer, collection, batch);
```

determine which of a large set of sto

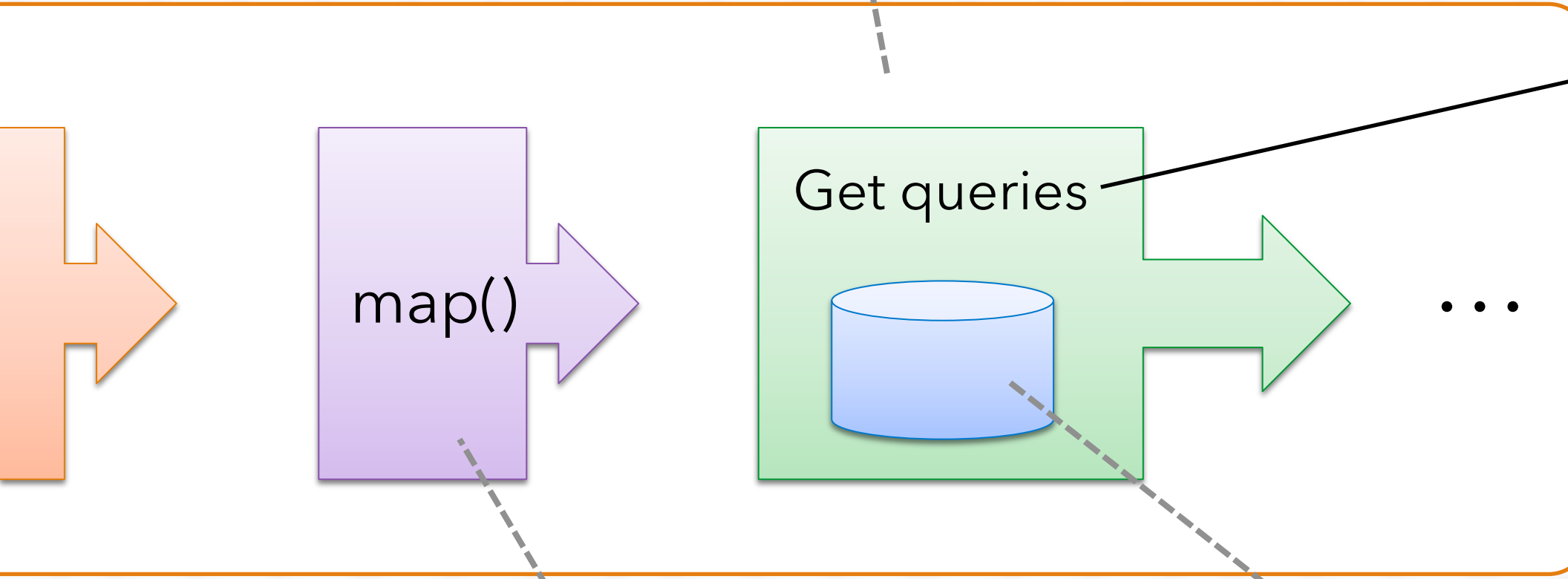
ernative flow paths through a stream,

into an embedded (in-memory) Solr
ch queries match

k; you have to decide where to load t
hat to do when matches are found

... need to scale to many queries, ch

JavaDStream<SolrInputDocument> enriched =
SolrSupport.*filterDocuments*(docFilterCon



Index docs into an
EmbeddedSolrServ
Initialized from conf
stored in ZooKeepe

m<SolrInputDocument> **docs** = **tweets**.**map**(

o each shard leader using metadata from
nment, and ConcurrentUpdateSolrClient

```
Partitioner = new ShardPartitioner(zkHost, col
tioner).foreachPartition(
Tuple2<String, SolrInputDocument>>>() {
r<Tuple2<String, SolrInputDocument>> tupleIter
ient cuss = null;
xt()) {
oncurrentUpdateSolrClient once per partition
```

ery and expose as an RDD

```
avaRDD<SolrDocument>
```

needed (cursorMark)

lt sets where global sort order doesn
ecution by distributing requests acros

```
nt> results =
```

struct RDD<Vector> which can then b

```
 SolrRDD(zkHost, collection);

rs =
ctors(jsc, solrQuery, field, numFeat


=
s.rdd(), numClusters, numIterations)
```

```
 SolrQuery(...);
_t","type_s");

rRDD(zkHost, collection);
rJavaRDD = solrRDD.queryShards(jsc, solrQuery
ew SQLContext(jsc);

Context, solrQuery, solrJavaRDD, zkHost, coll

ets");


COUNT(type_s) FROM tweets WHERE type_s='echo'

s.javaRDD().map(new Function<Row, Long>() {
```

ntation of Solr and Spark on YARN

for reads and writes to Solr

- improving replication performance

to Solr Scale Toolkit

give visibility into performance

to me with questions: