
Spark

Java?

`pyspark.sql.DataFrame.count`

```
    def count(self) -> int:                                                [docs]
        """Returns the number of rows in this :class:`DataFrame`.

        .. versionadded:: 1.3.0

        Examples
        --------
        >>> df.count()
        2
        """
        return int(self._jdf.count())
```

Java?!

`pyspark.sql.DataFrame.collect`

```
    def collect(self) -> List[Row]:                                    [docs]
        """Returns all the records as a list of :class:`Row`.

        .. versionadded:: 1.3.0

        Examples
        --------
        >>> df.collect()
        [Row(age=2, name='Alice'), Row(age=5, name='Bob')]
        """
```

Java?!

`pyspark.sql.DataFrame.collect`

```
    def collect(self) -> List[Row]:                                    [docs]
        """Returns all the records as a list of :class:`Row`.

        .. versionadded:: 1.3.0

        Examples
        --------
        >>> df.collect()
        [Row(age=2, name='Alice'), Row(age=5, name='Bob')]
        """
        with SCCallSiteSync(self._sc):
            sock_info = self._jdf.collectToPython()
        return list(_load_from_socket(sock_info, BatchedSerializer(CPickleSerializer())))
```

*“**Pickling**” is the process whereby a Python object hierarchy is converted into a byte stream, and “**unpickling**” is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy*

- python docs

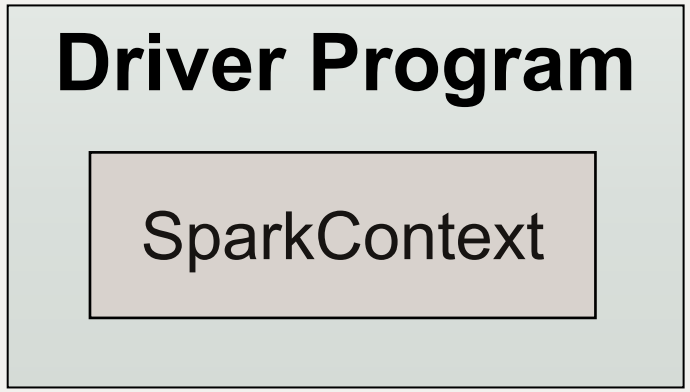
What if…

functional programming…

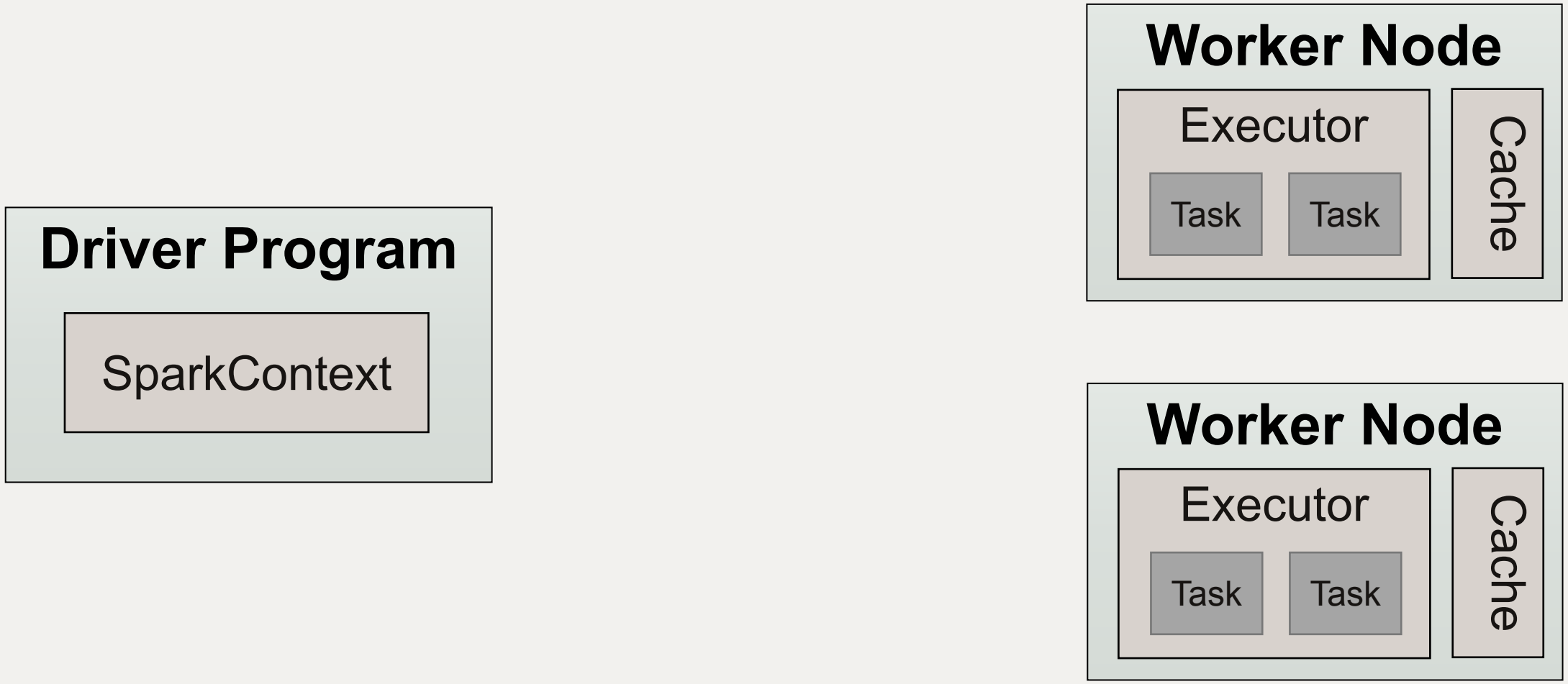
on data…

and distributed!

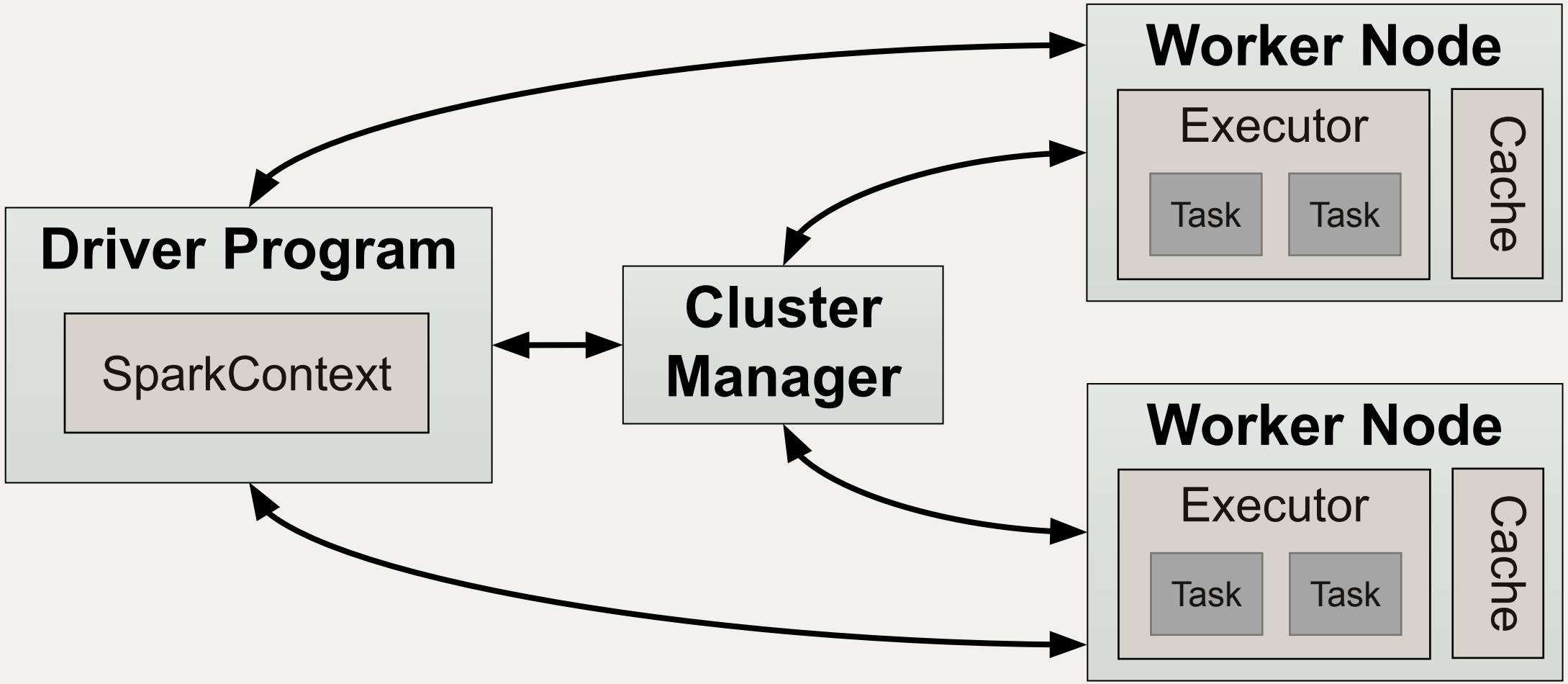
Distribution!



Distribution!



Distribution!



Show the dashboard

Because Matei Zaharia wanted

to do ***MapReduce*** *but better*

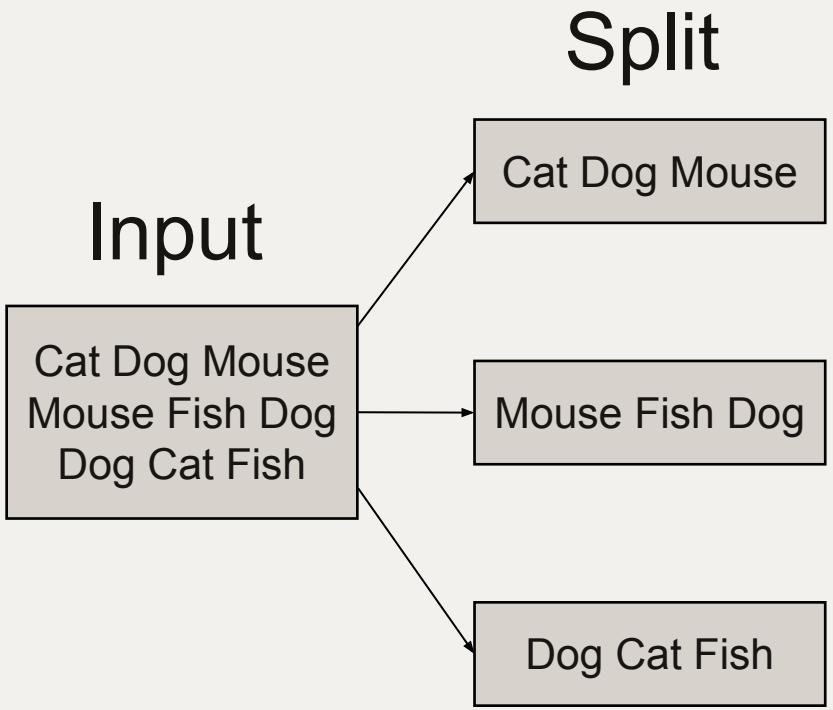
Let’s count words

Let’s count words

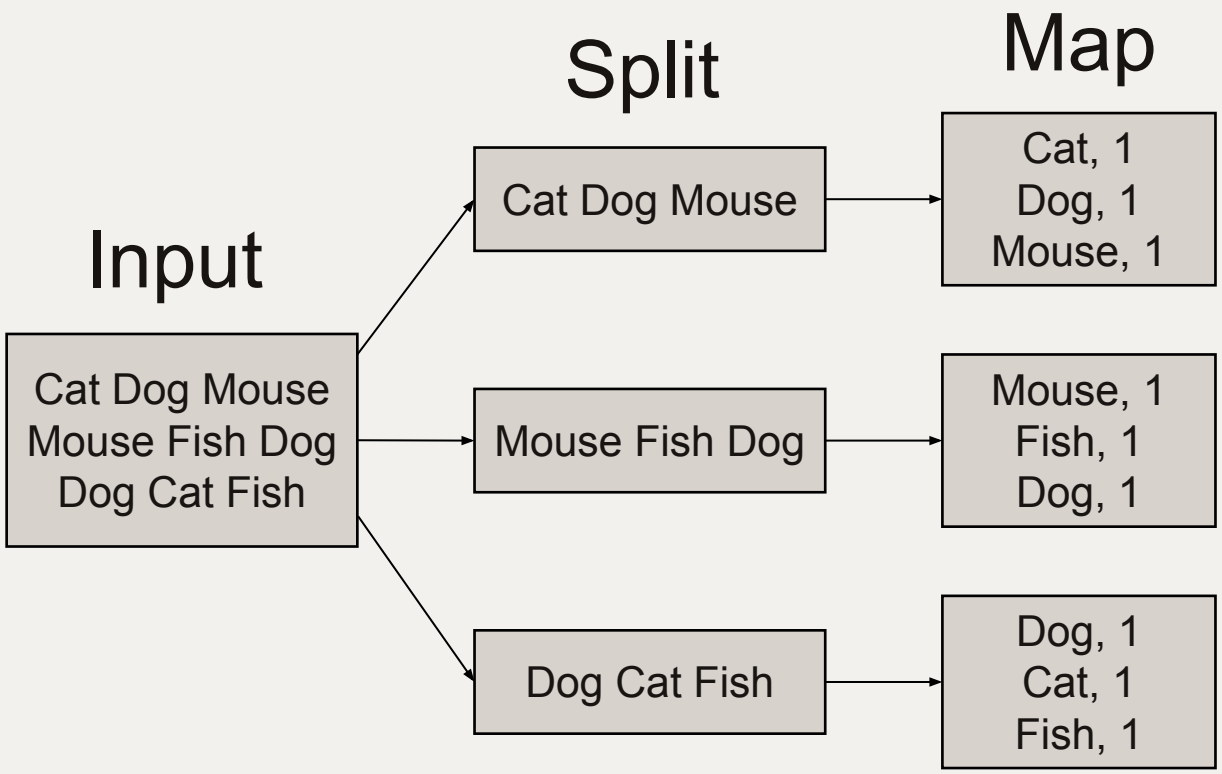
Input

```
Cat Dog Mouse
Mouse Fish Dog
Dog Cat Fish
```

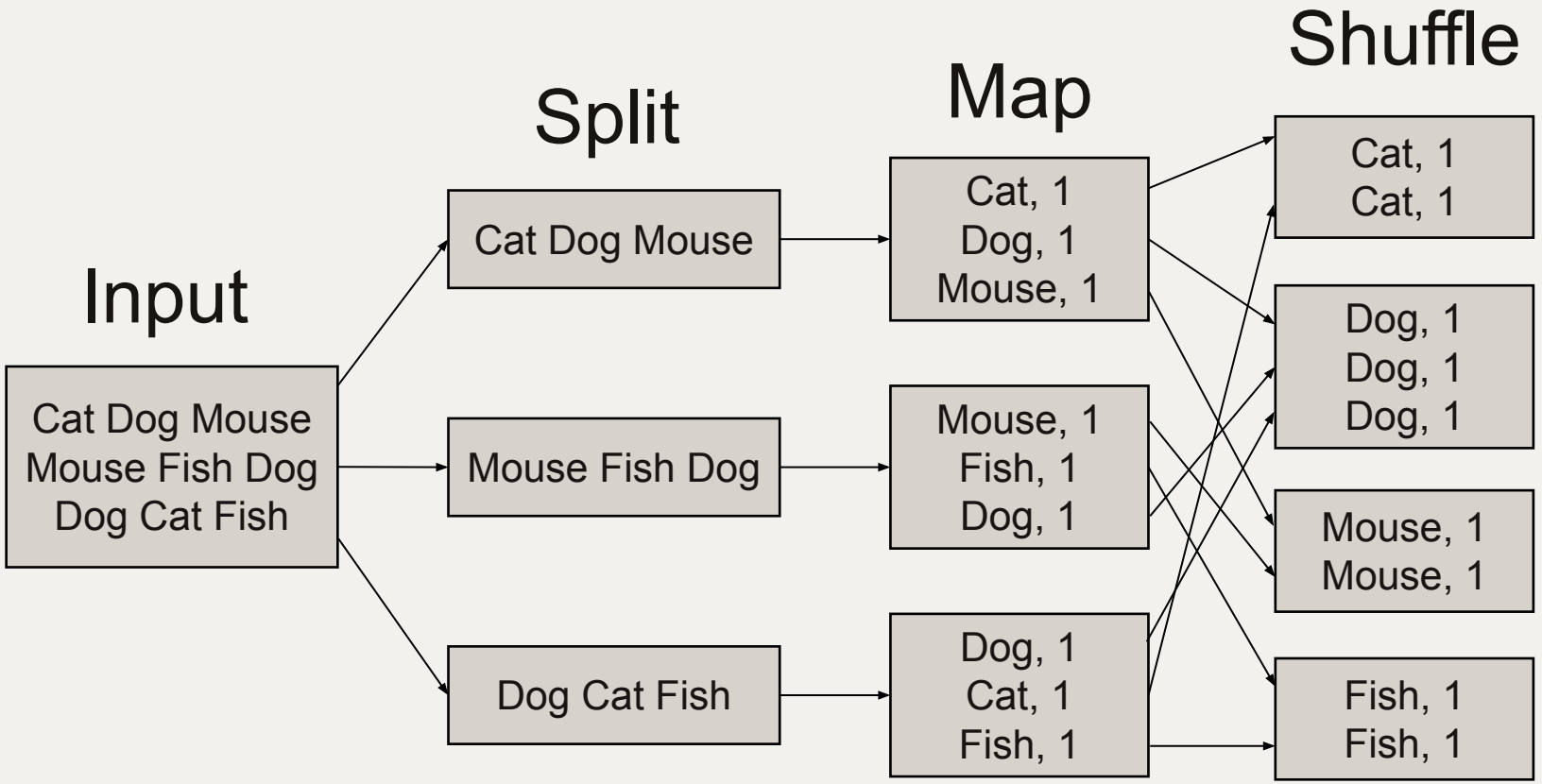
Let’s count words



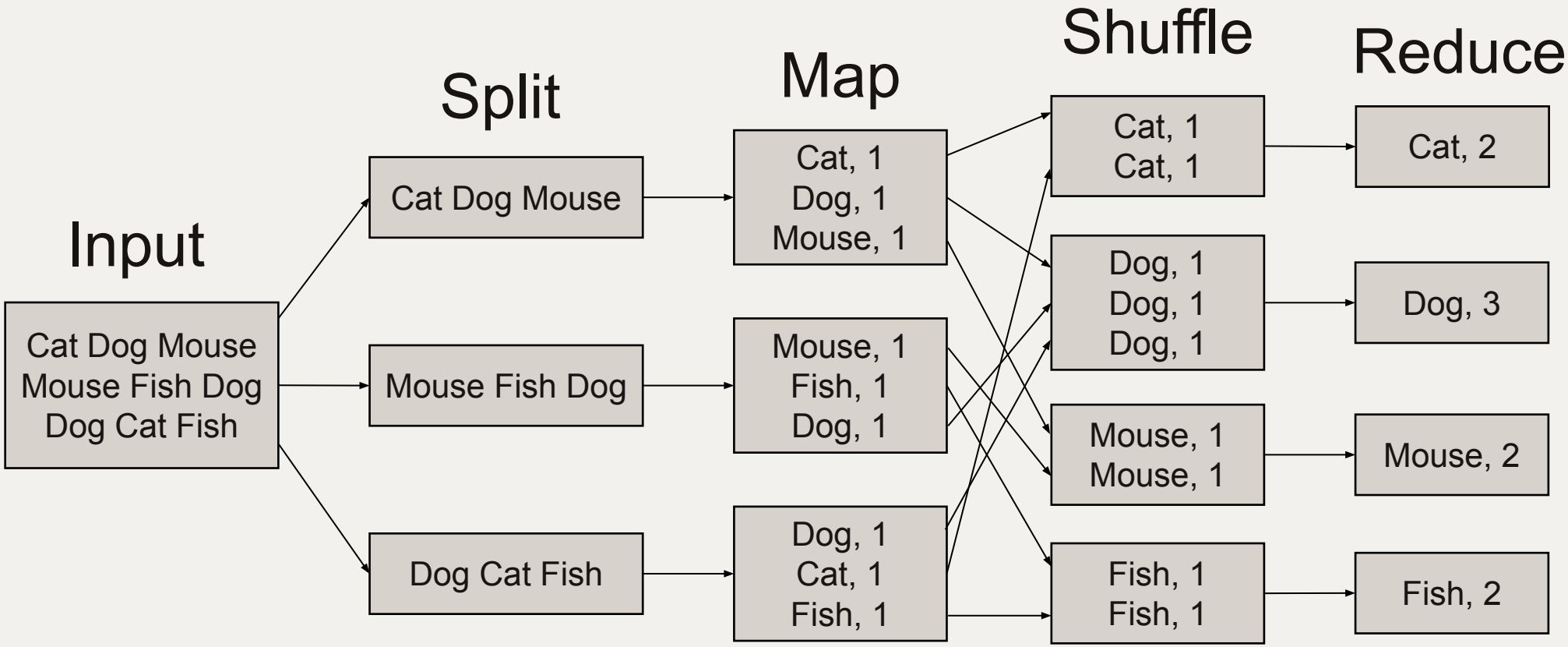
Let’s count words



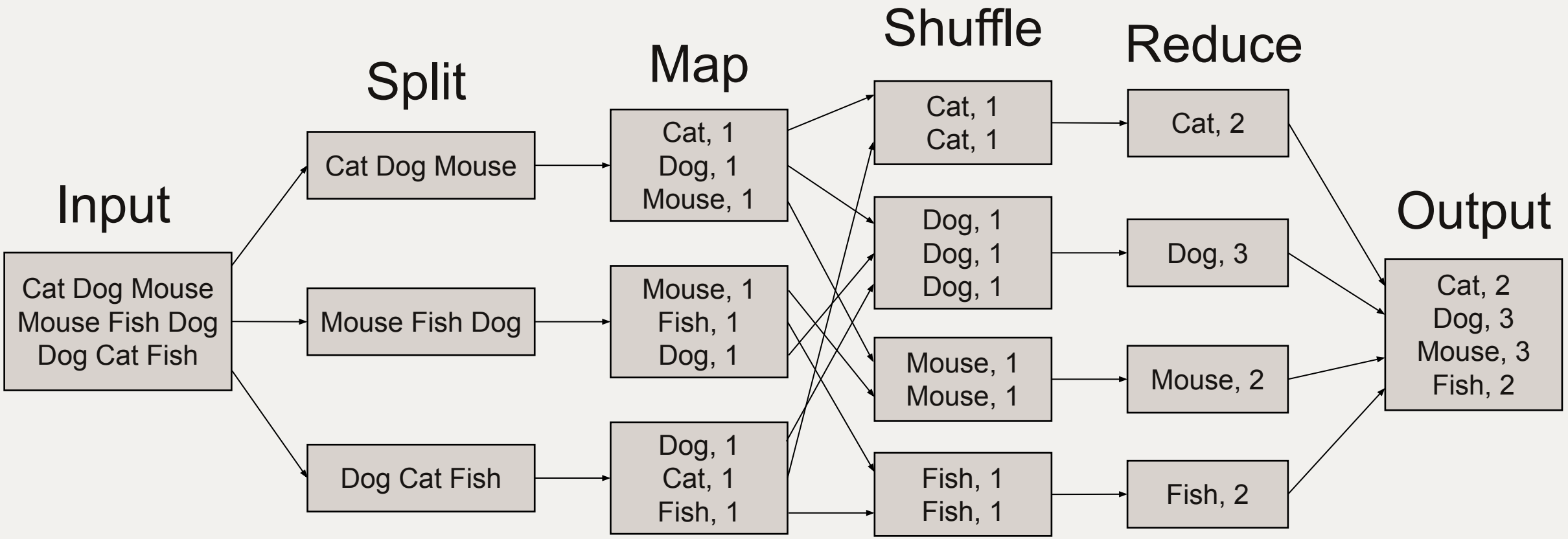
Let’s count words



Let’s count words

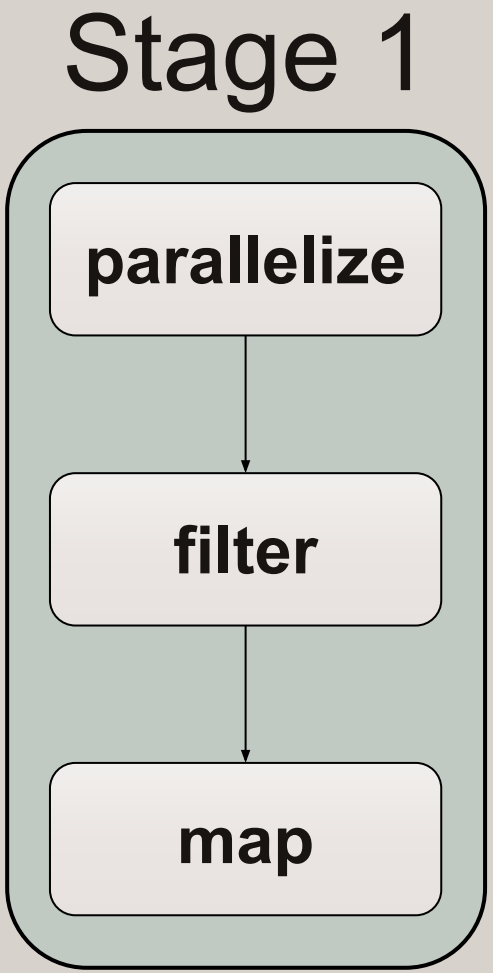


Let’s count words

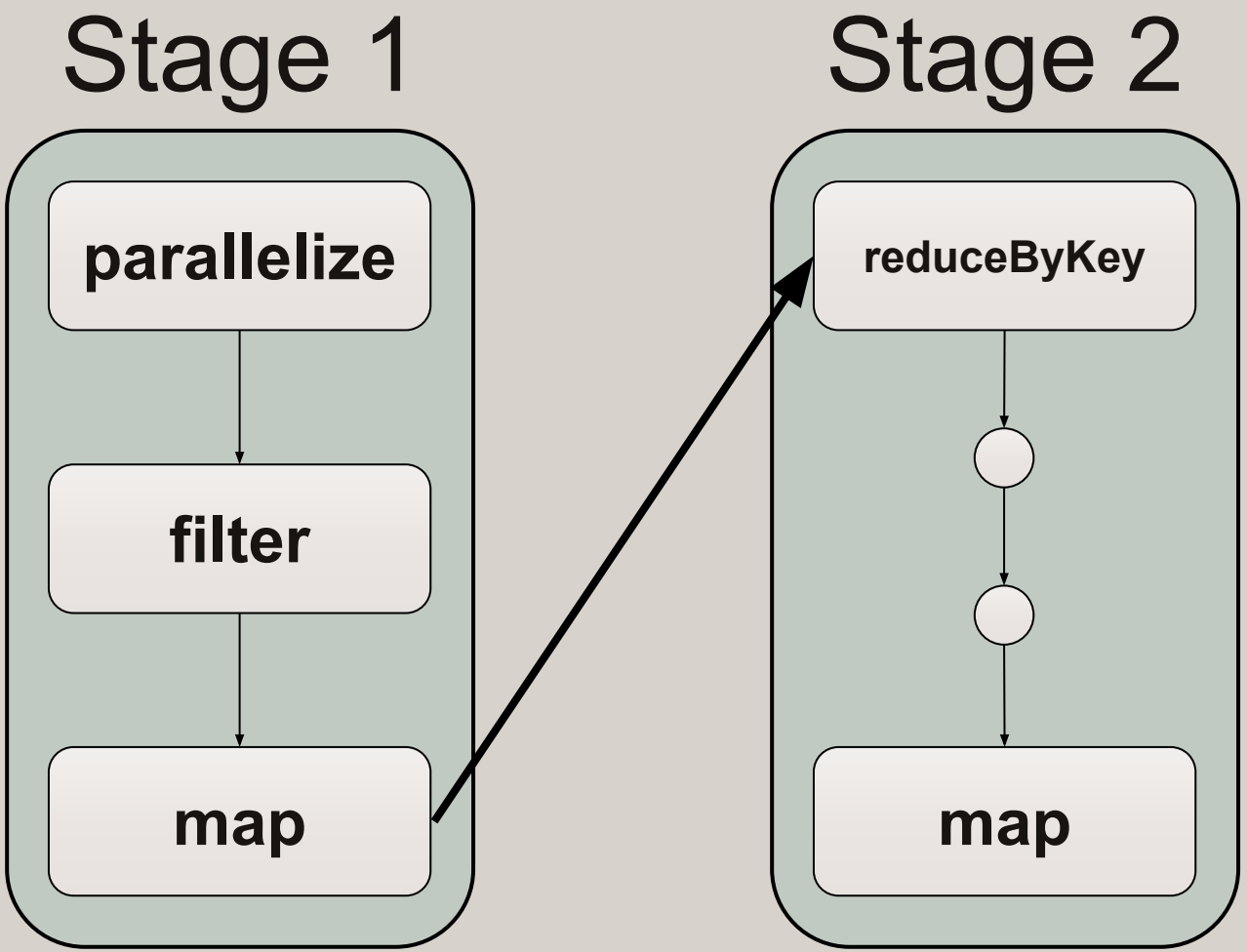


Directed Acyclic Graphs

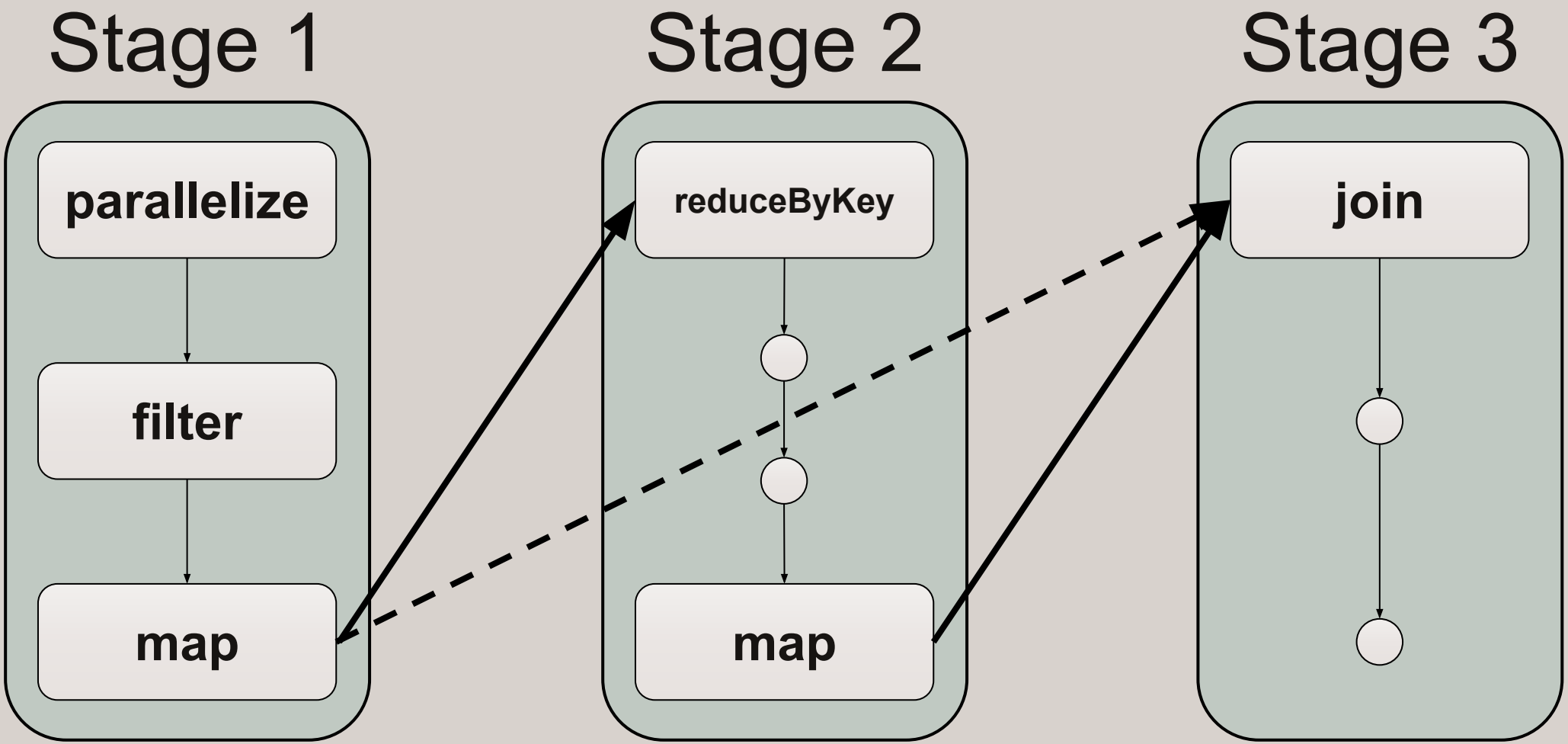
Directed Acyclic Graphs



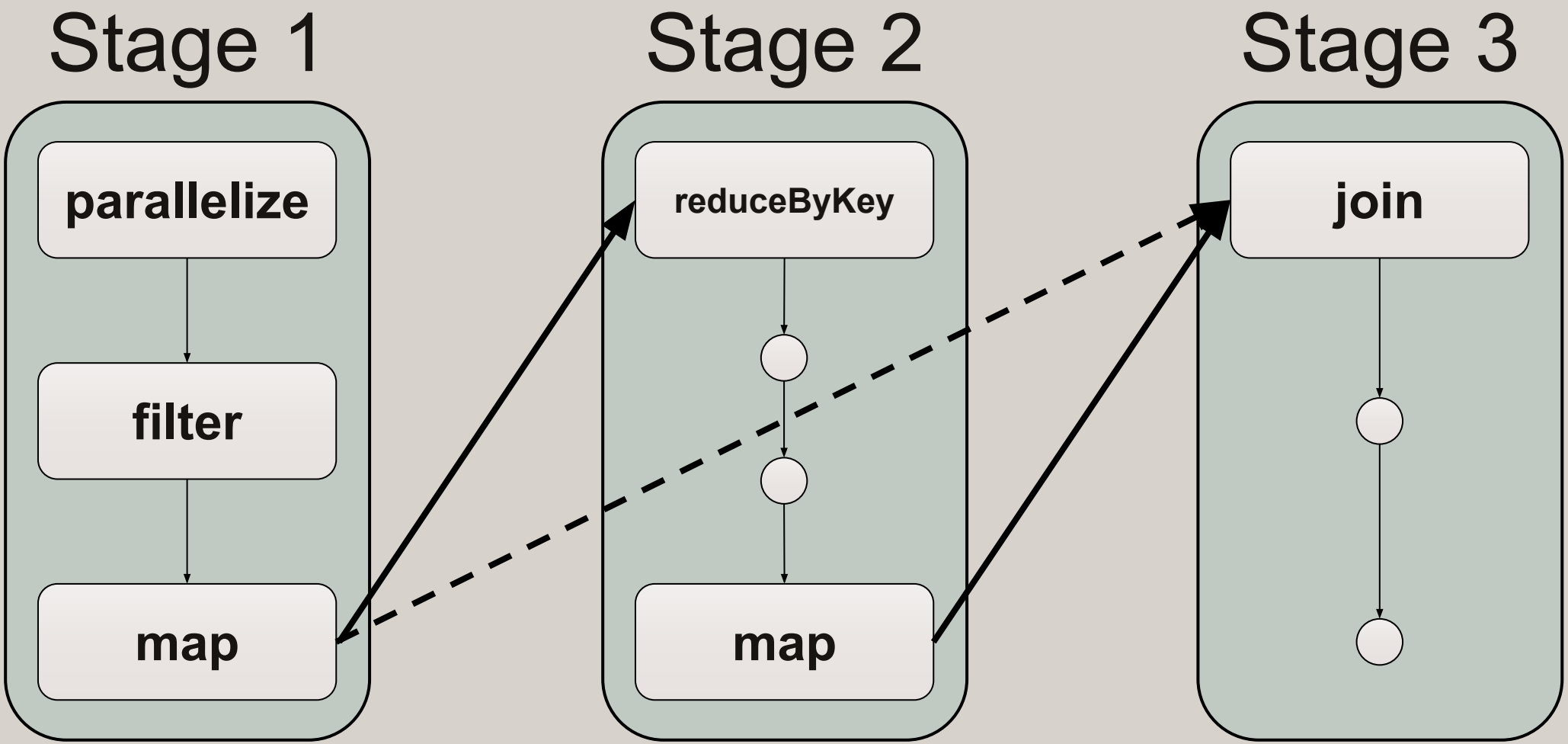
Directed Acyclic Graphs



Directed Acyclic Graphs

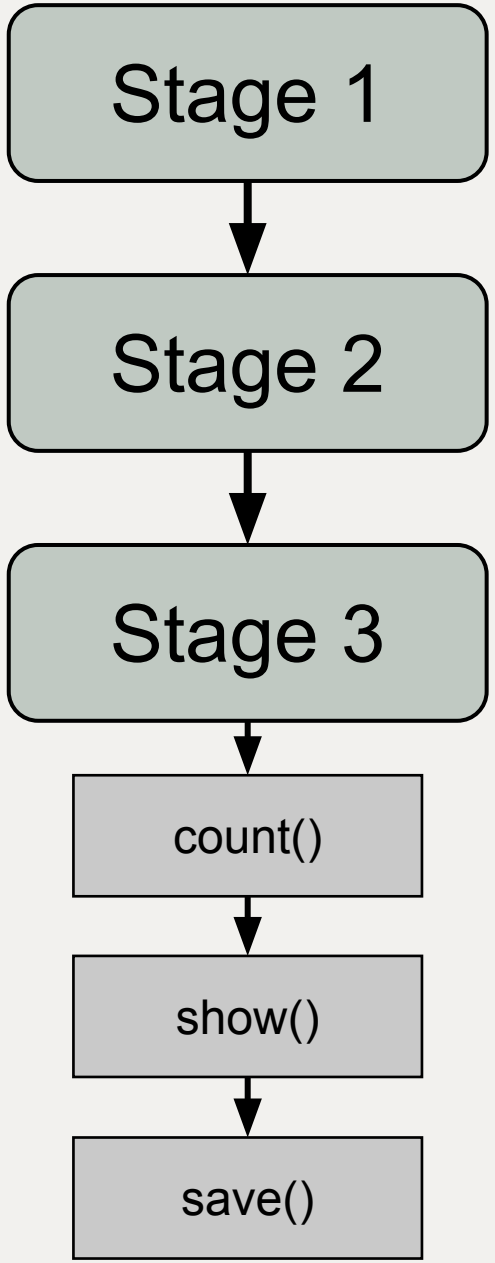


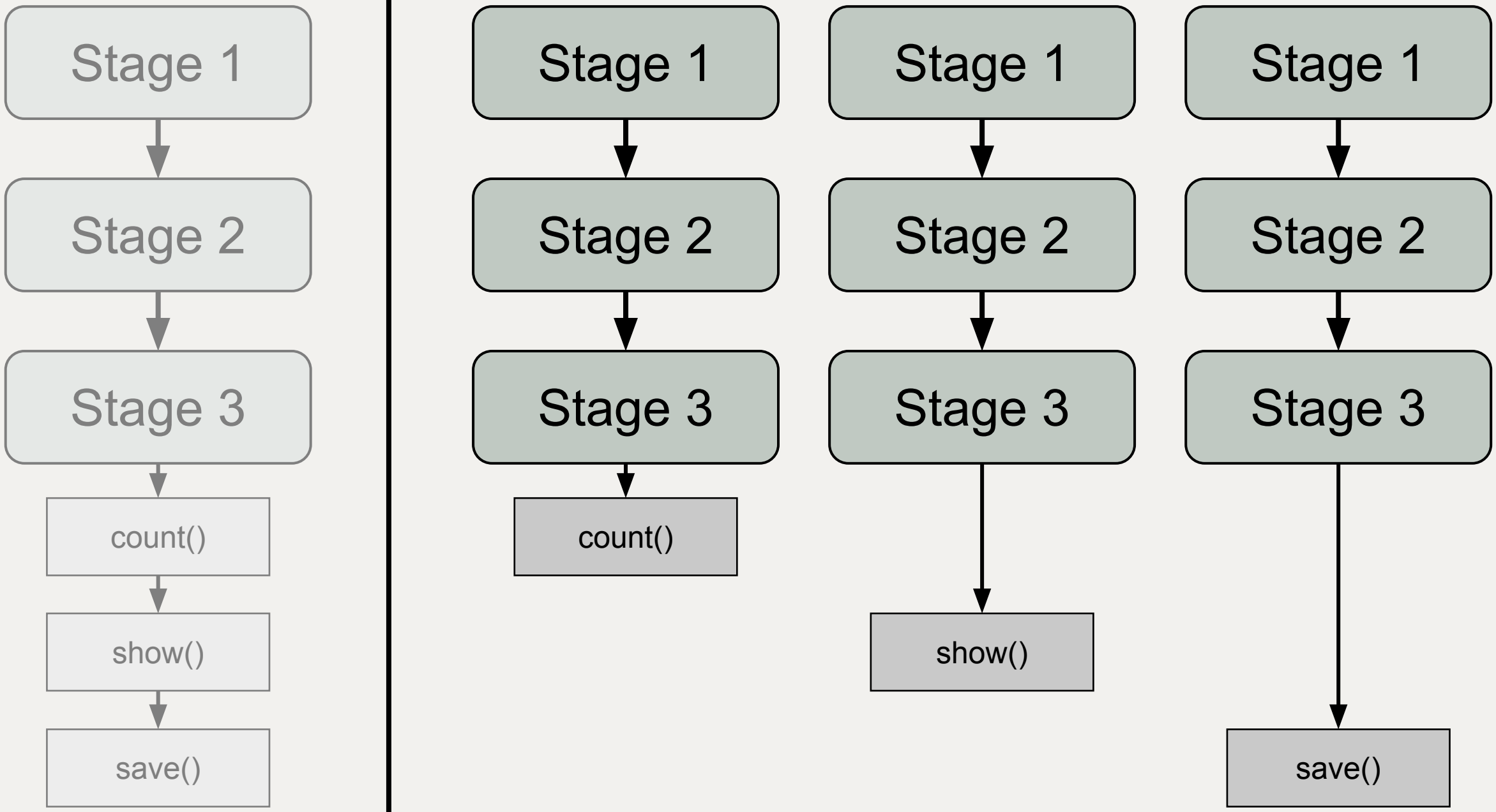
java.lang.OutOfMemoryError



Transformations *- lazy, added to the plan, generally take a dataframe and return a dataframe*

Actions *- eager, force evaluation of the plan*


Stage 1
Stage 2
Stage 3
count()
show()
save()


Stage 1
Stage 2
Stage 3
count()
show()
save()
Stage 1
Stage 2
Stage 3
count()
Stage 1
Stage 2
Stage 3
show()
Stage 1
Stage 2
Stage 3
save()

Memory Management

Cache/persist

- *Lazy*
- *Maintains lineage*
- *Choose storage level*

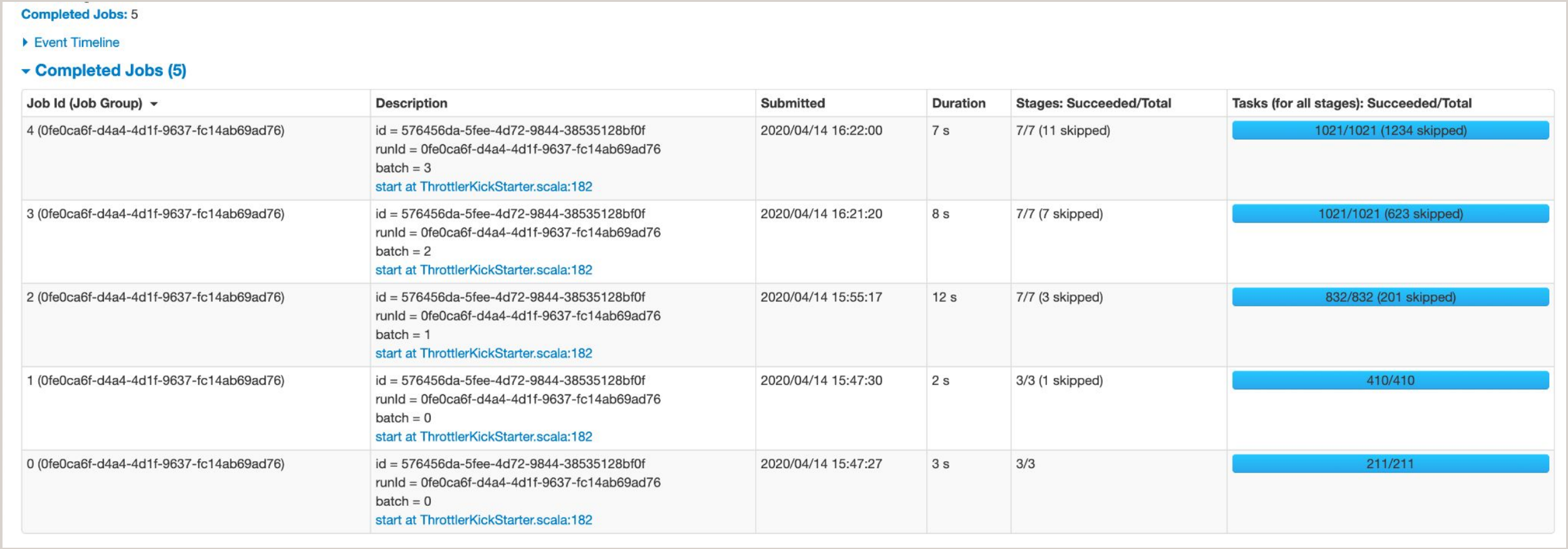
Checkpoint

- *Lazy or eager*
- *Destroys lineage*
- *Maintained after spark application terminated*

Save as …

- *(e.g. saveAsTable)*

Memory Management



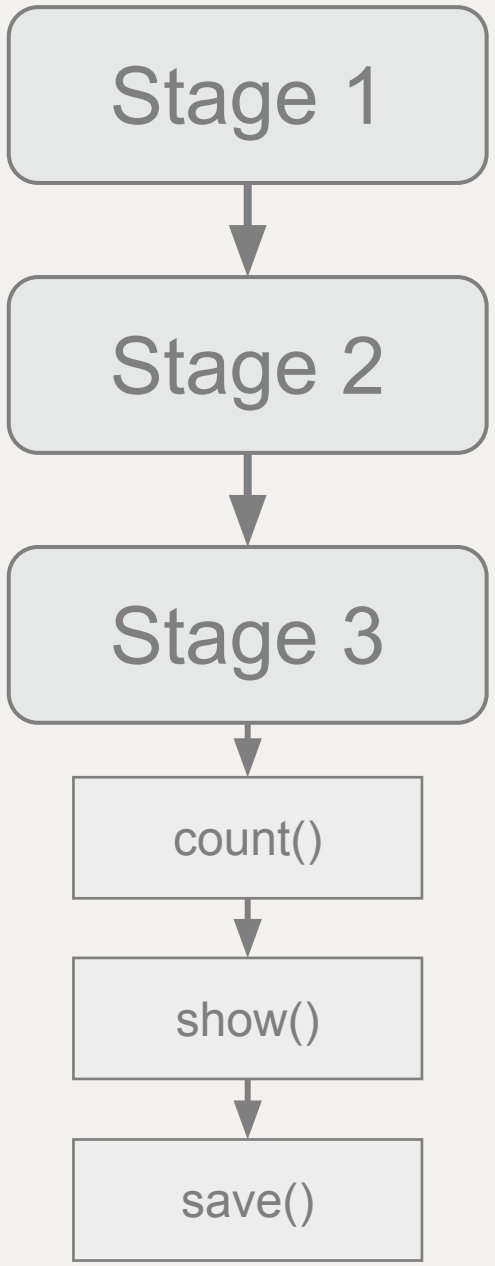
Completed Jobs: 5

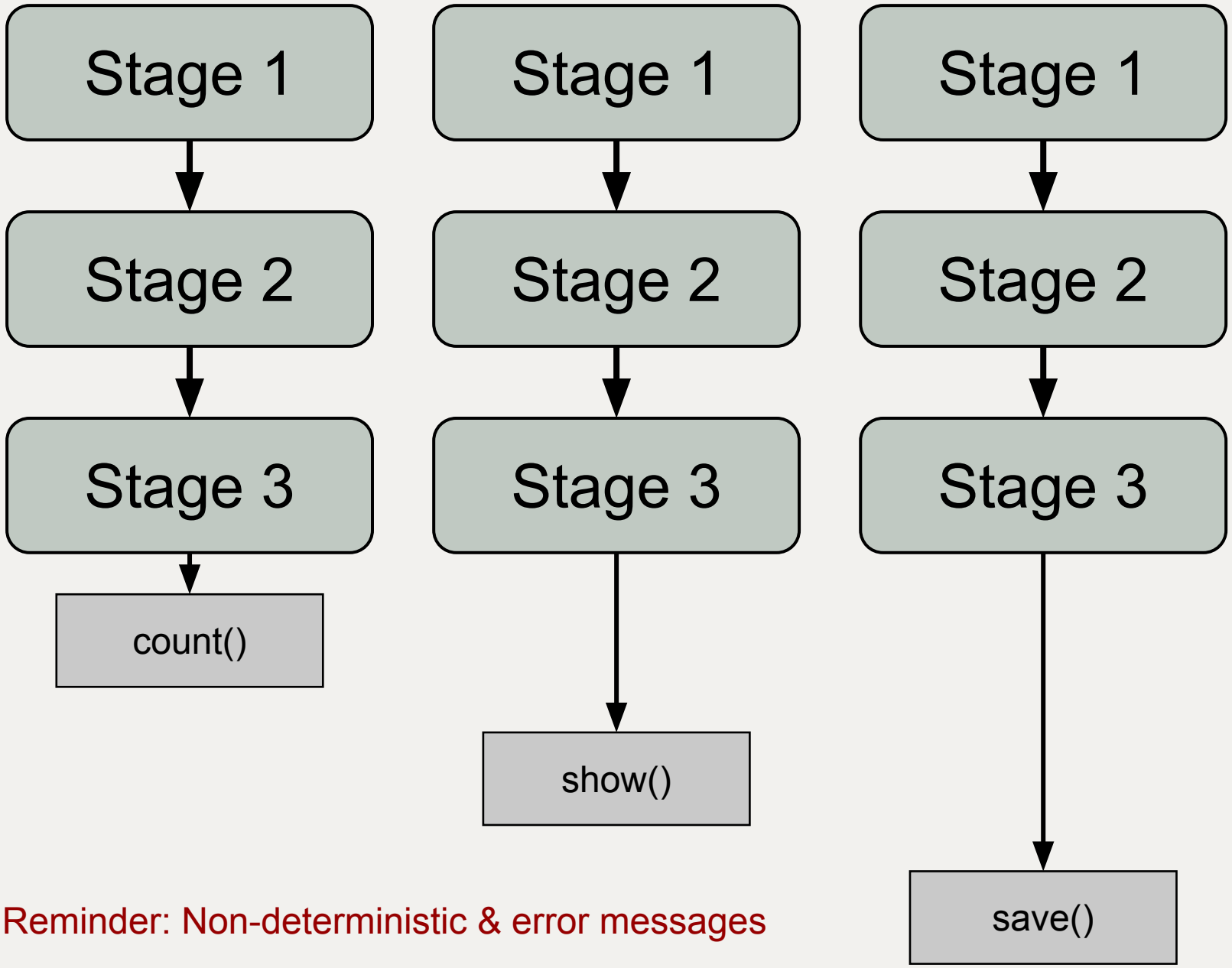
▸ Event Timeline

▾ **Completed Jobs (5)**

Job Id (Job Group)	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
4 (0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76)	id = 576456da-5fee-4d72-9844-38535128bf0f runId = 0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76 batch = 3 start at ThrottlerKickStarter.scala:182	2020/04/14 16:22:00	7 s	7/7 (11 skipped)	1021/1021 (1234 skipped)
3 (0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76)	id = 576456da-5fee-4d72-9844-38535128bf0f runId = 0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76 batch = 2 start at ThrottlerKickStarter.scala:182	2020/04/14 16:21:20	8 s	7/7 (7 skipped)	1021/1021 (623 skipped)
2 (0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76)	id = 576456da-5fee-4d72-9844-38535128bf0f runId = 0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76 batch = 1 start at ThrottlerKickStarter.scala:182	2020/04/14 15:55:17	12 s	7/7 (3 skipped)	832/832 (201 skipped)
1 (0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76)	id = 576456da-5fee-4d72-9844-38535128bf0f runId = 0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76 batch = 0 start at ThrottlerKickStarter.scala:182	2020/04/14 15:47:30	2 s	3/3 (1 skipped)	410/410
0 (0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76)	id = 576456da-5fee-4d72-9844-38535128bf0f runId = 0fe0ca6f-d4a4-4d1f-9637-fc14ab69ad76 batch = 0 start at ThrottlerKickStarter.scala:182	2020/04/14 15:47:27	3 s	3/3	211/211

https://stackoverflow.com/questions/61206084/does-skipped-stages-have-any-performance-impact-on-spark-job

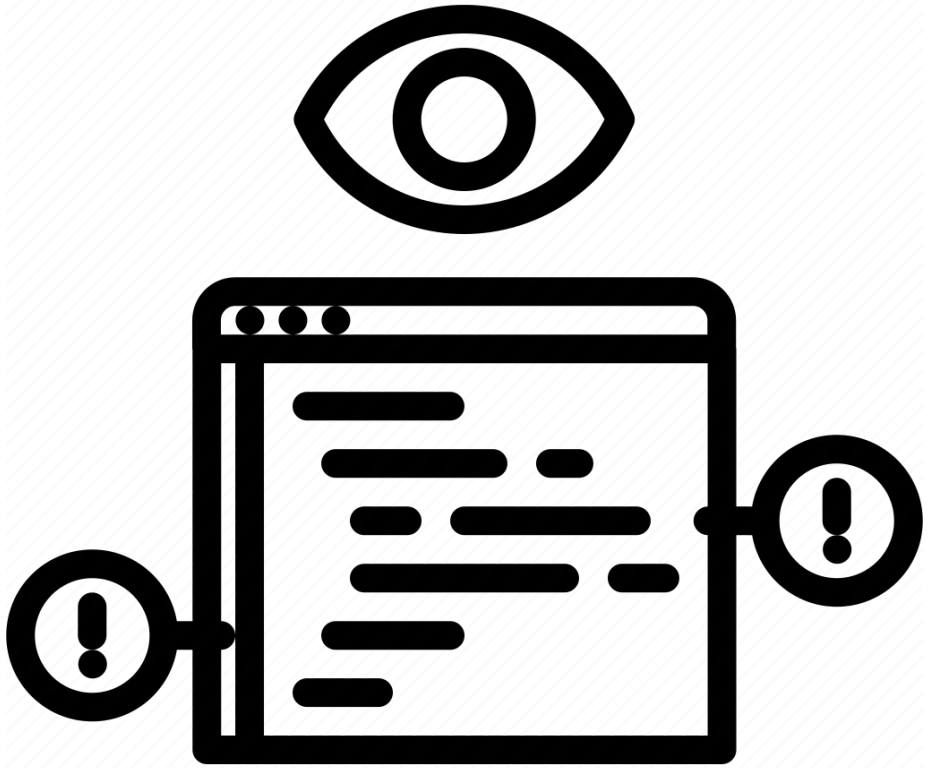




Reminder: Non-deterministic & error messages

```
Py4JJavaError: An error occurred while calling o793.count.
: org.apache.spark.SparkException: Job aborted due to stage failure: Task 0 in stage 423.0 failed 1 times, most recent failure: Lost task 0.0 in stage
423.0 (TID 15778) (IT187328.emea.porsche.biz executor driver): org.apache.spark.SparkException: Python worker failed to connect back.
        at org.apache.spark.api.python.PythonWorkerFactory.createSimpleWorker(PythonWorkerFactory.scala:188)
        at org.apache.spark.api.python.PythonWorkerFactory.create(PythonWorkerFactory.scala:108)
```

https://stackoverflow.com/questions/70151751/what-does-this-error-message-mean-in-pyspark

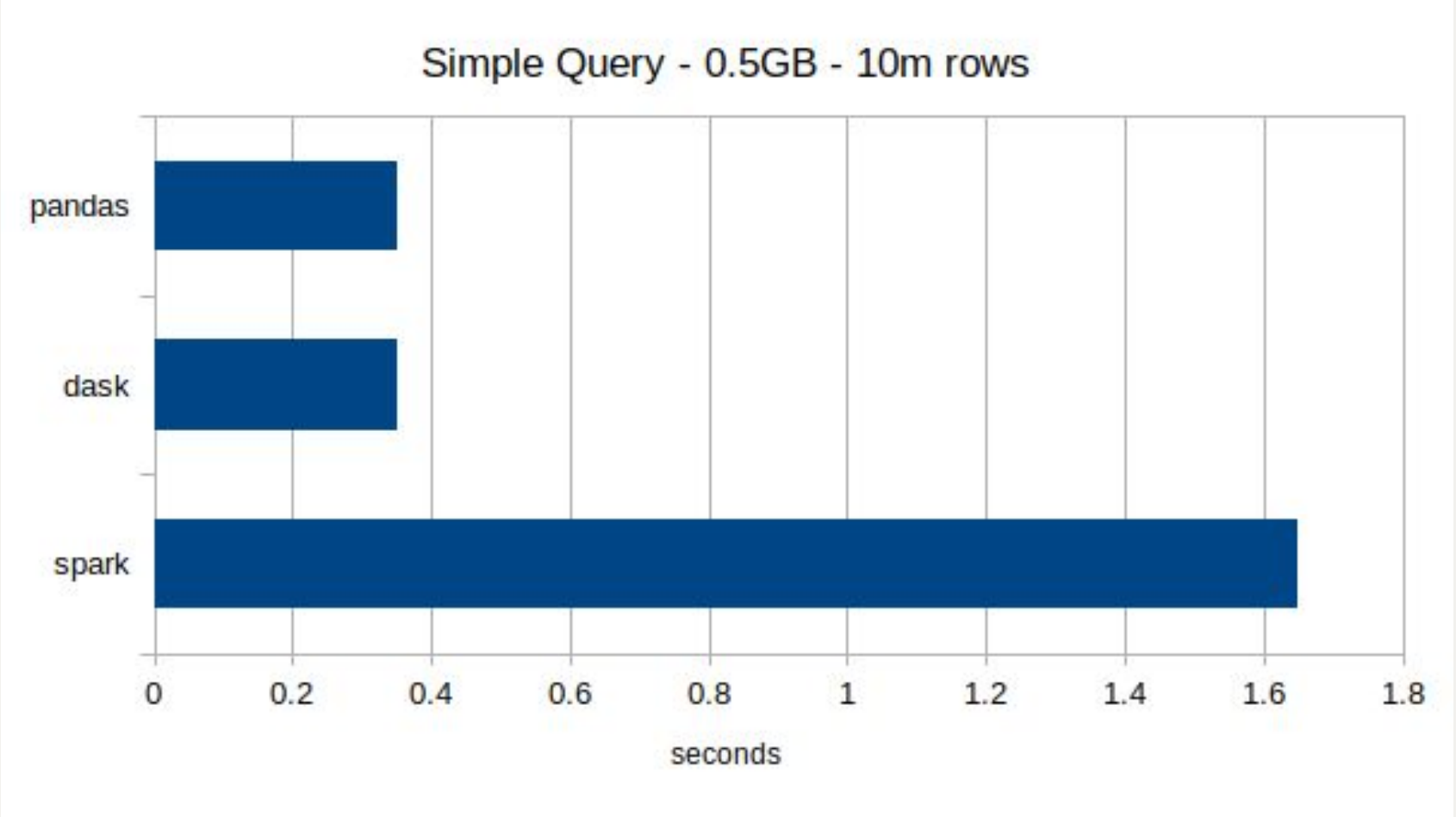


What if you had not one, but four Java stack traces?

So what should I use?

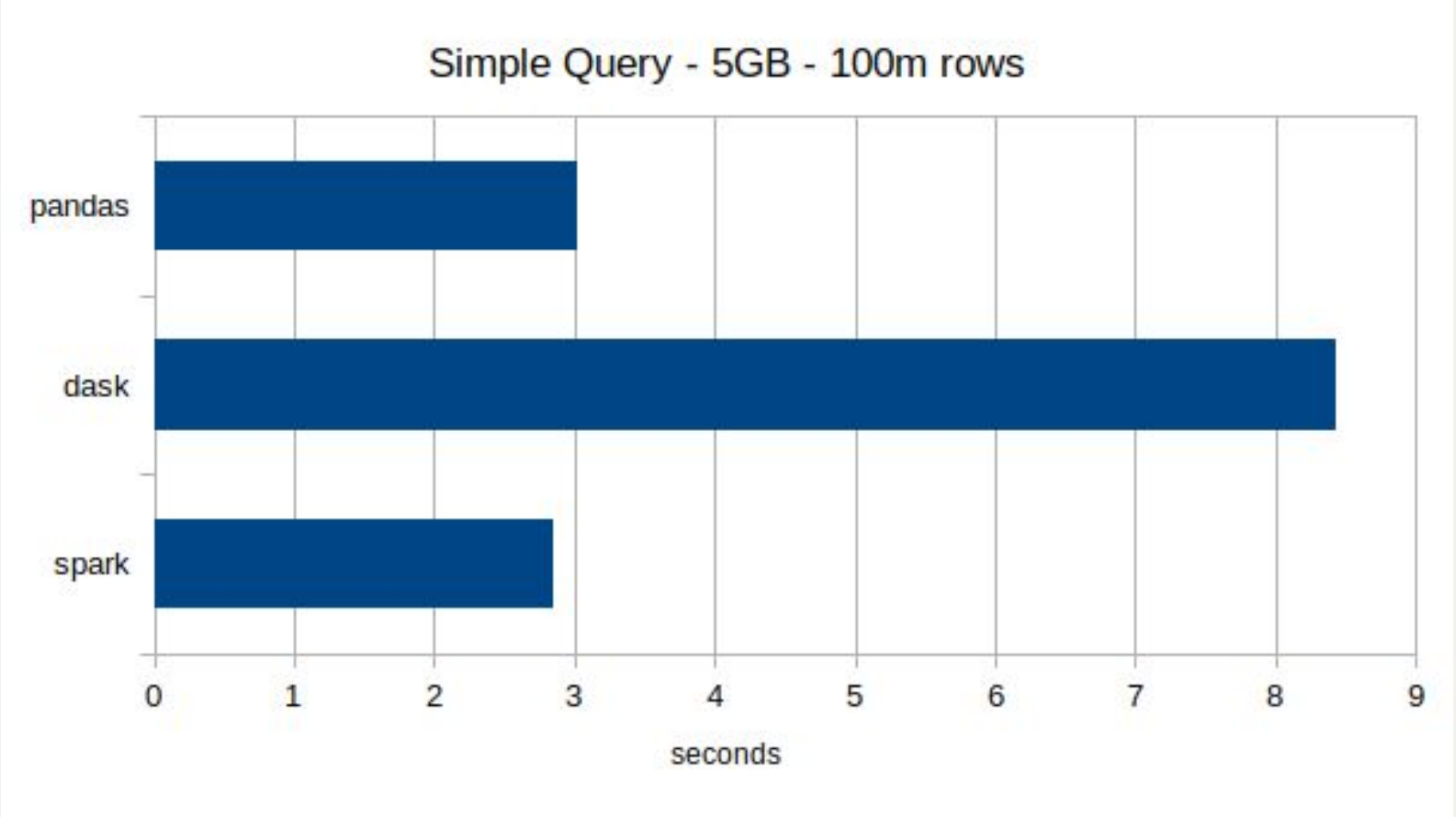
- How much data do you have?
- What does your current code base look like?
- How much time do you want to spend on infrastructure?
- Who is working on your code?
- Any personal preferences?

So what should I use?



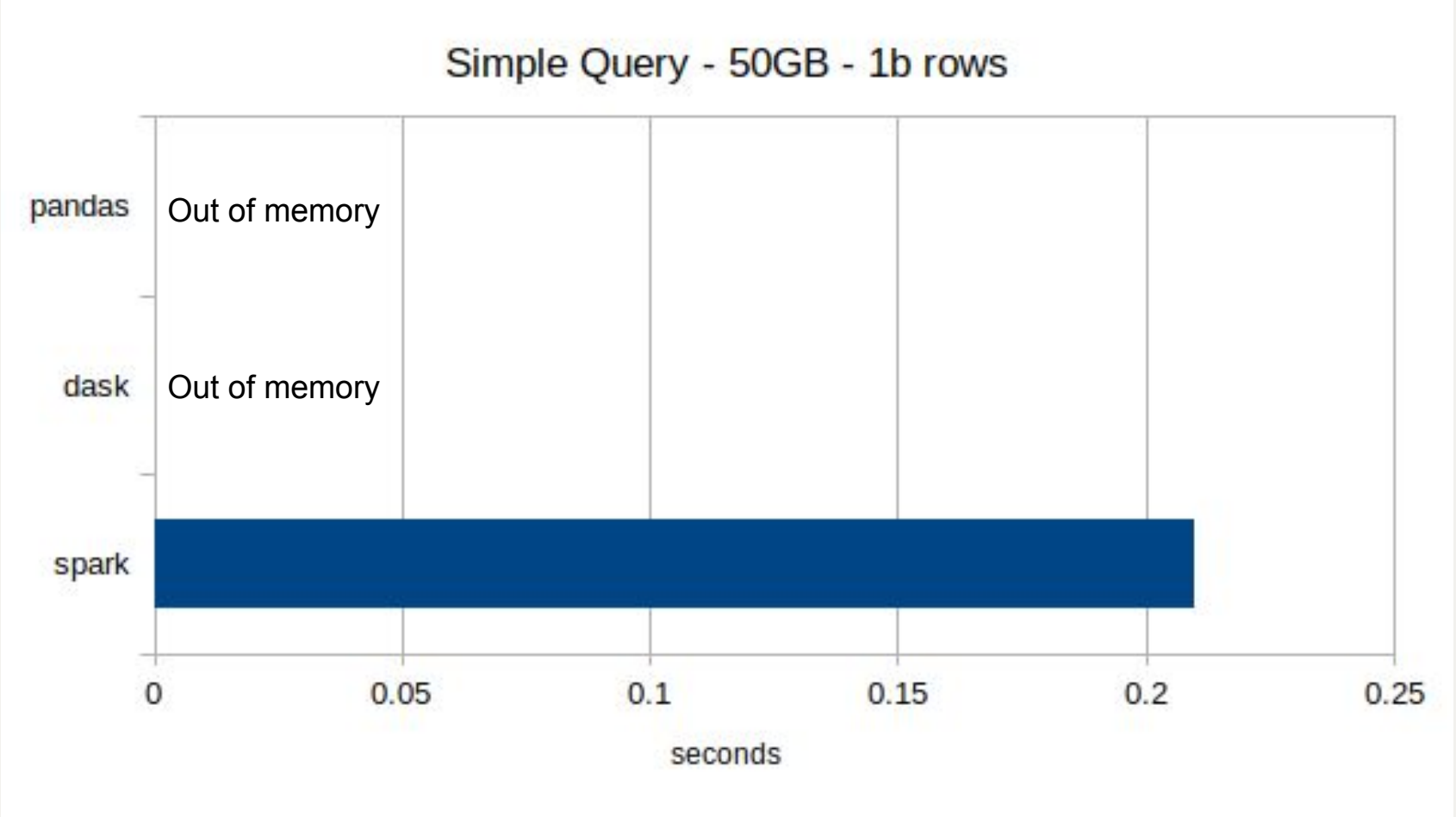
https://h2oai.github.io/db-benchmark/

So what should I use?



https://h2oai.github.io/db-benchmark/

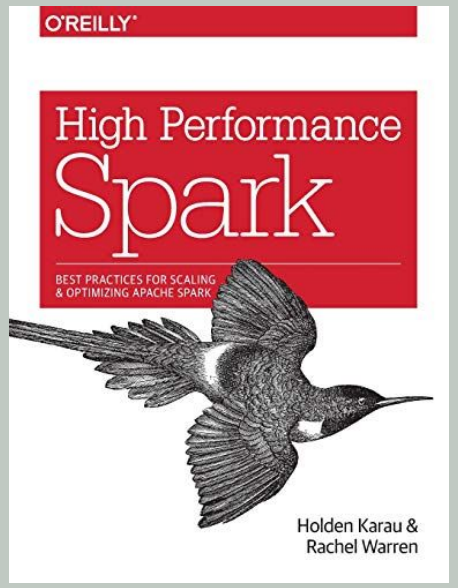
So what should I use?

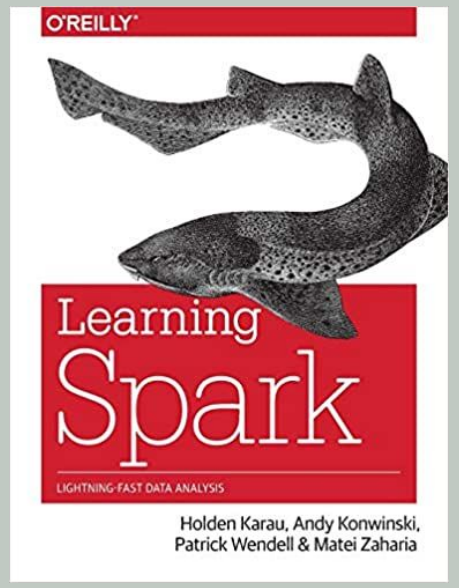


https://h2oai.github.io/db-benchmark/

Holden Karau

Holden is a transgender Canadian open source developer with a focus on Apache Spark, Airflow, Kubeflow, and related “big data” tools. She is the co-author of Learning Spark, High Performance Spark, and Kubeflow for Machine Learning. She is a committer and PMC on Apache Spark.
https://databricks.com/speaker/holden-karau





Improving PySpark Performance: Spark performance beyond the JVM
https://youtu.be/jGhju2bw3RQ

In Conclusion

- What is PySpark?
- Can it solve all of my data problems?
- Are you sure I can’t just use pandas instead?

In Conclusion

- What is PySpark?
  See previous slides
- Can it solve all of my data problems?
- Are you sure I can’t just use pandas instead?

In Conclusion

- What is PySpark?
  See previous slides
- Can it solve all of my data problems?
  Kind of, but you'll also get fun new ones
- Are you sure I can't just use pandas instead?

In Conclusion

- What is PySpark?
  See previous slides
- Can it solve all of my data problems?
  Kind of, but you'll also get fun new ones
- Are you sure I can't just use pandas instead?

Other Stuff

- Deploy mode (client vs cluster)
- Data skew (salting method)
- RDDs vs DataFrames vs Datasets
- User defined functions (UDFs)
- Pandas, streaming and machine learning
- Setting it all up (*https://phoenixnap.com/kb/install-spark-on-ubuntu*)

Commands

```
cd /opt/spark/sbin

./start-master.sh

./start-worker.sh spark://{url}

pyspark

    df = spark.createDataFrame([("a",1),("b",2)], schema=("val","id"))

    df.show()

./stop-worker.sh

./stop-all.sh
```