This slide is blank on purpose

Riding Jet Streams

http://bit.ly/streams_jfokus2017

@gAmUssA @hazelcast #jfokus #hazelcastjet

> WHOAMI

Solutions Architect @Hazelcast

Developer Advocate @Hazelcast

@gamussa in internetz

Please, follow me on Twitter
I'm very interesting ©

Agenda

Quick refresh on Java 8 Streams
Distribute and Conquer
Distributed Data
Distributed Streams
How we did all this

Example: Word Count

Map<Integer, String> where keys are line numbers and values are lines.

Find how many times each word occurs

What needs to be done?

Iterate through all the lines
Split the line into words
Update running total of counts with new word

Iterate through all the lines

```
fillMapWithData("war_and_peace_eng.txt", source);

for (String line : source.values()) {
    for (String word : PATTERN.split(line)) {
    if (word.length() >= 5)
      count.compute(
          cleanWord(word).toLowerCase(),
          (w, c) -> c == null ? 1 : c + 1
      );
  }
}
System.out.println(count.get("andrew"));
```

Split the line into words

```
fillMapWithData("war_and_peace_eng.txt", source);

for (String line : source.values()) {
    for (String word : PATTERN.split(line)) {
    if (word.length() >= 5)
      count.compute(
          cleanWord(word).toLowerCase(),
          (w, c) -> c == null ? 1 : c + 1
      );
  }
}
System.out.println(count.get("andrew"));
```

Update running total of counts with new word

```
fillMapWithData("war_and_peace_eng.txt", source);

for (String line : source.values()) {
    for (String word : PATTERN.split(line)) {
    if(word.length() >= 5)
      count.compute(
          cleanWord(word).toLowerCase(),
          (w, c) -> c == null ? 1 : c + 1
      );
  }
}
System.out.println(count.get("andrew"));
```

Print the result

```
fillMapWithData("war_and_peace_eng.txt", source);

for (String line : source.values()) {
    for (String word : PATTERN.split(line)) {
    if(word.length()>=5)
      count.compute(
          cleanWord(word).toLowerCase(),
          (w, c)->c == null ? 1 : c + 1
      );
  }
}
System.out.println(count.get("andrew"));
```

JAVA.UTIL.STREAM

Java 8 Streams...

An abstraction represents a sequence of elements
Is not a data structure
Convey elements from a source through a pipeline of operations
Operation doesn't modify a source

Why I should care about Stream API?

You're Java developer

What does ADVANCED ~~REGULAR~~ Java developer think about Scala?

What is Scala


Martin Odersky

- 纯面向对象,class\trait\mixin
- 函数式first class,lambda,closure,curry,lazy,tail recursive opt
- Actor, pattern-match
- Jvm bytecode(1.5 compatible)
- 强类型,静态语言,

Why I should care about Stream API?

You're Java developer
Many Java developers know Java
It's all about data processing

java.util.stream

Intermediate operation

`map(), flatMap(), filter()`

Terminal operation

`reduce(), collect()`

Stateful Intermediate (Blocking) operation

`sorted(), distinct()`


FRESH CUT POTATOES
COOKED IN 100% PEANUT OIL
NO CHOLESTEROL OR PRESERVATIVES
100% FRESH AMERICAN BEEF
NEVER FROZEN
NO FILLERS OR PRESERVATIVES
100% FRESH AMERICAN BEEF NEVER FROZEN
NO FILLERS OR PRESERVATIVES
FAMOUS BURGERS
HAMBURGER 4.39
CHEESEBURGER 4.99
BACON BURGER 5.09
BACON CHEESEBURGER 5.69
FAMOUS BURGERS
LITTLE HAMBURGER 3.29
LITTLE CHEESEBURGER 3.69
LITTLE BACON BURGER 3.99
LITTLE BACON CHEESEBURGER 4.29
FAMOUS DOGS
KOSHER STYLE HOT DOG 2.89
CHEESE or BACON DOG 3.09
BACON CHEESE DOG 3.89
VEGGIE or GRILLED CHEESE 2.79
FAMOUS FRIES
REGULAR 2.49
LARGE 4.19
REGULAR 1.79
LARGE 1.89
BOTTLED WATER 1.79
ALL TOPPINGS FREE
Mayo Relish Onions Lettuce Pickles Tomatoes Grilled Onions Grilled Mushrooms Ketchup Mustard
Jalapeno Peppers Green Peppers A-1 Sauce Bar-B-Q Sauce Hot Sauce "UPON REQUEST ONLY"
FRESH CUT POTATOES
COOKED IN 100% PEANUT OIL
NO CHOLESTEROL OR PRESERVATIVES
WE COOK ALL OUR MEAT JUICY AND WELL-DONE.
FIVE GUYS
Reynolds


FIVE GUY
BURGERS and FRI
FIVE GU


TALK IS CHEAP
SHOW ME THE CODE
memegenerator.net

Why would one need a cluster?

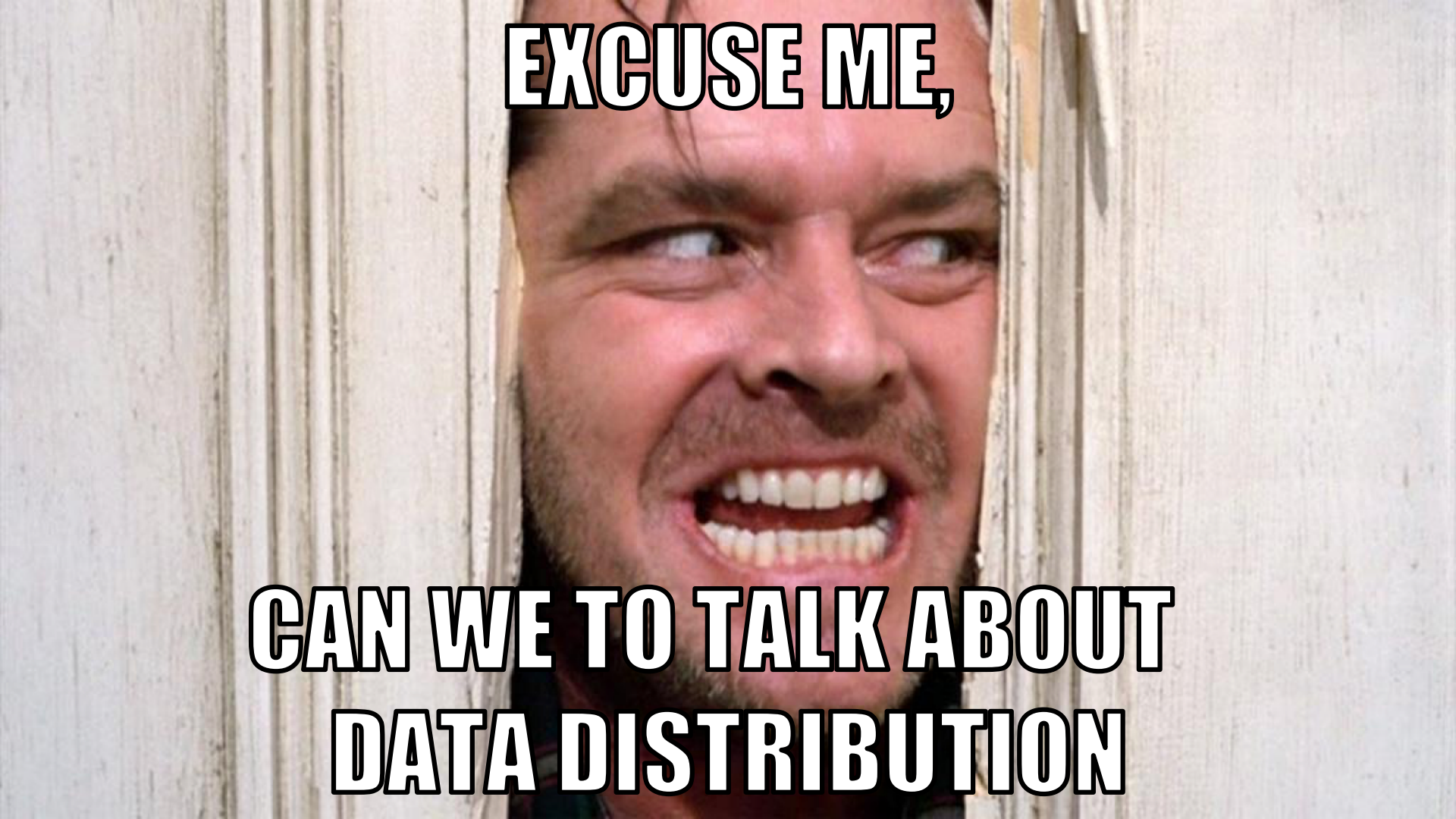
One does not simply fit all **Big Data** in one machine


ONE DOES NOT SIMPLY
FIT BIG DATA IN ONE MACHINE

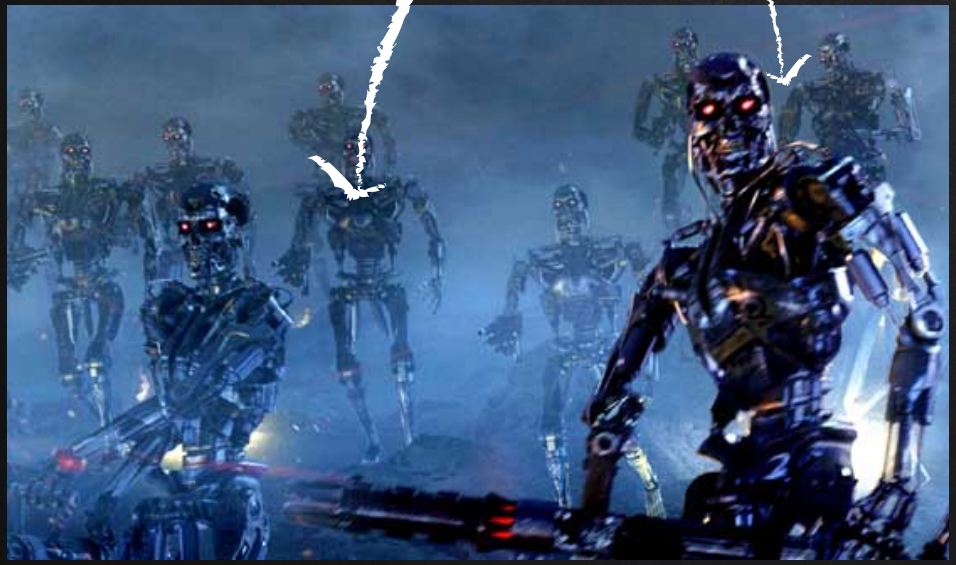
Why would one need a cluster?

One does not simply put all **Big Data** in one machine

Data is too important to have it only one machine

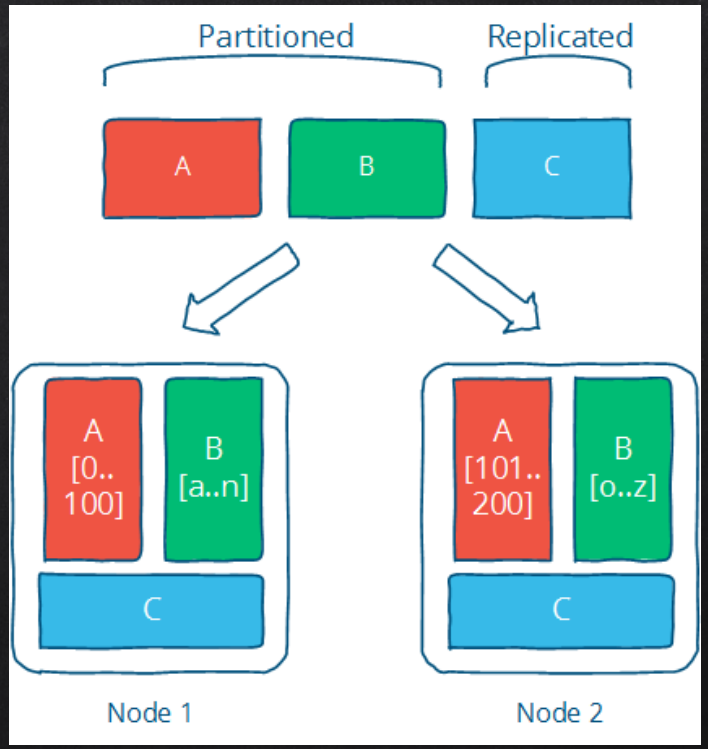

EXCUSE ME,
CAN WE TO TALK ABOUT
DATA DISTRIBUTION

REPLICATION



SHARDING

Replication on Sharding?



http://book.mixu.net/distsys/single-page.html

Another Requirements

Easy to use
Simple API
Embeddable
Cloud Native

hazelcast IMDG

What's Hazelcast IMDG?

In-memory Data Grid

Apache v2 Licensed

Distributed

Caches (IMap, JCache)

Java Collections (IList, ISet, IQueue)

Messaging (Topic, RingBuffer)

Computation (ExecutorService, M-R)

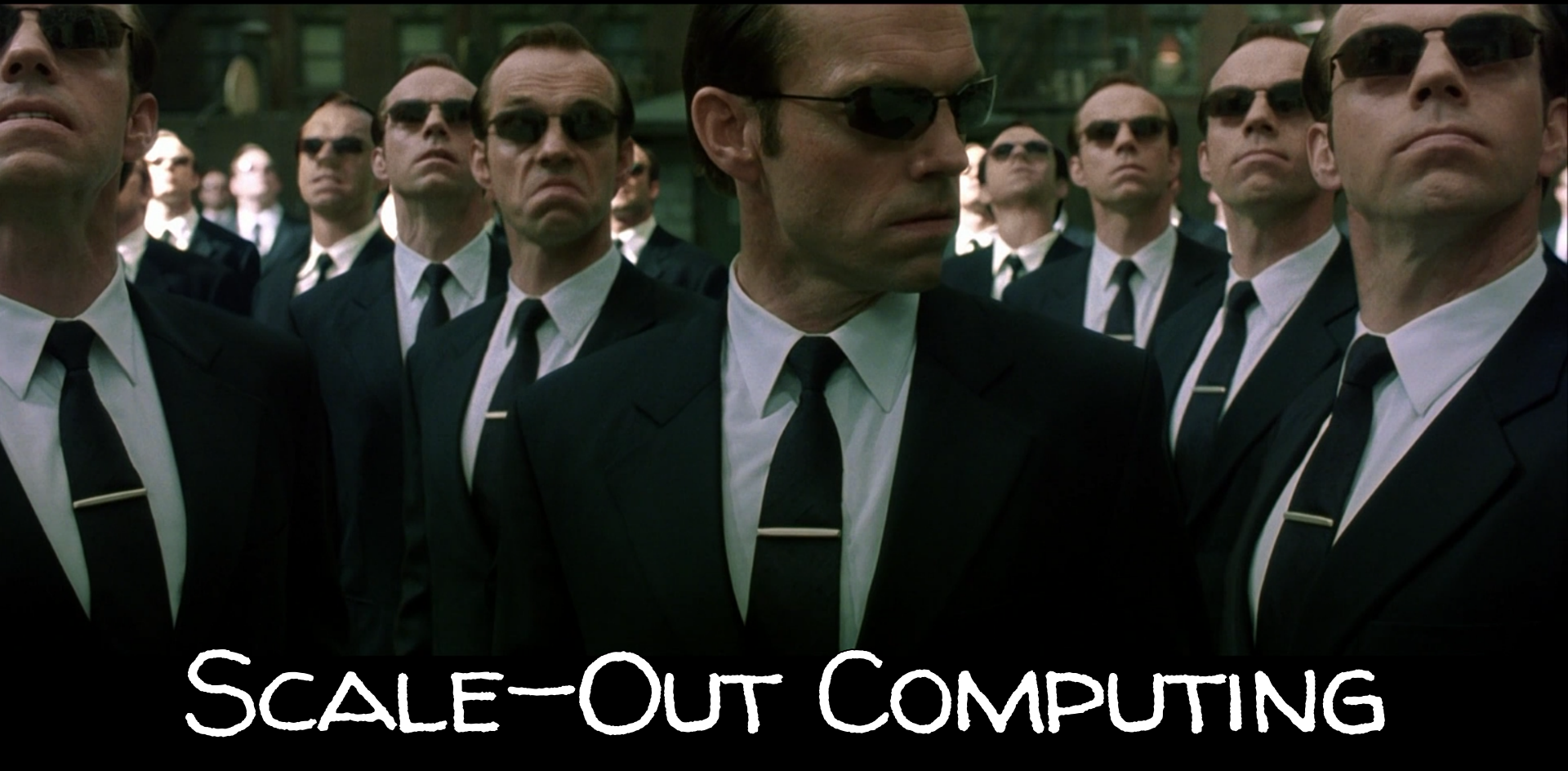
1 900 Stars

On GitHub

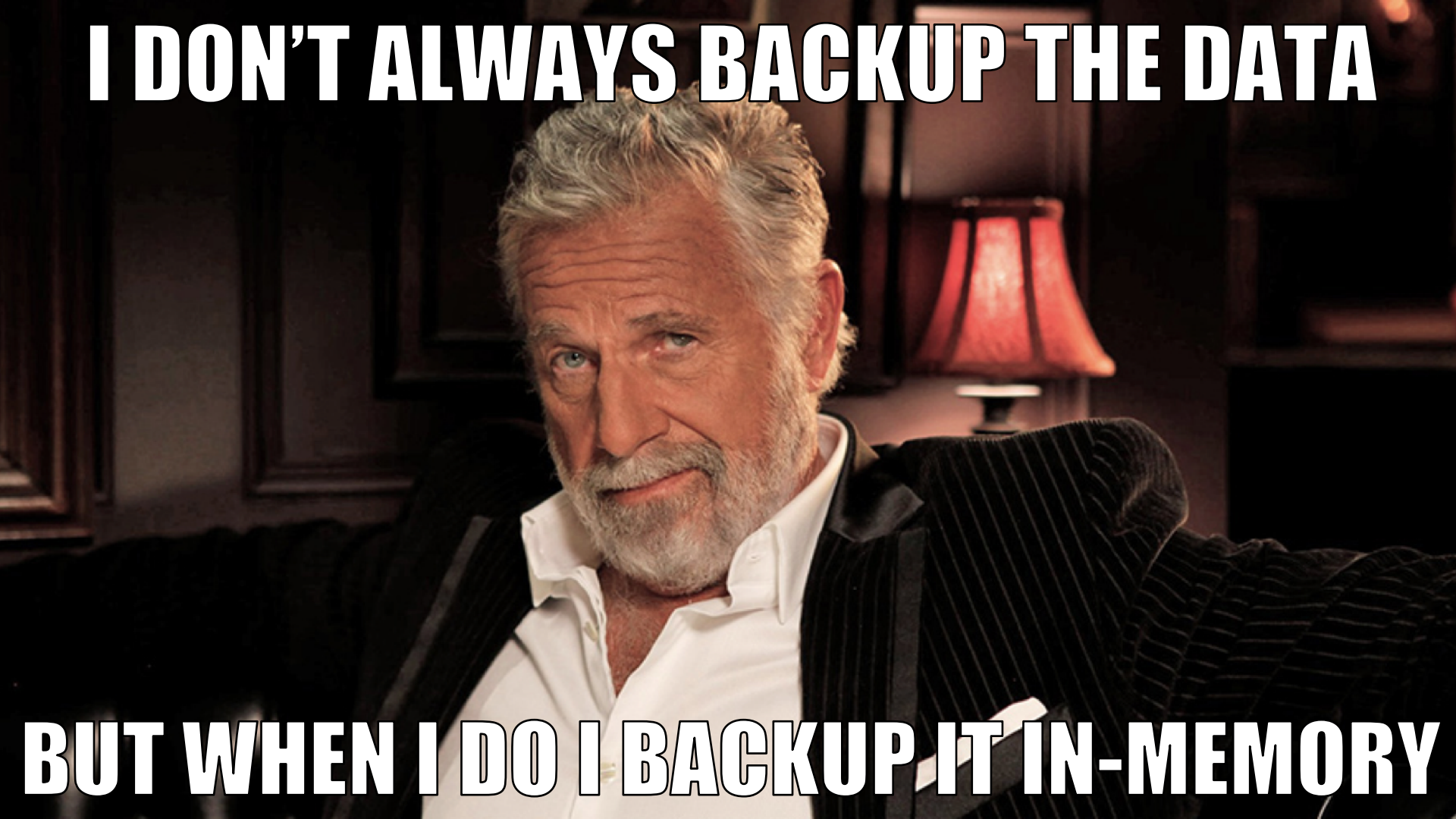
134 contributors

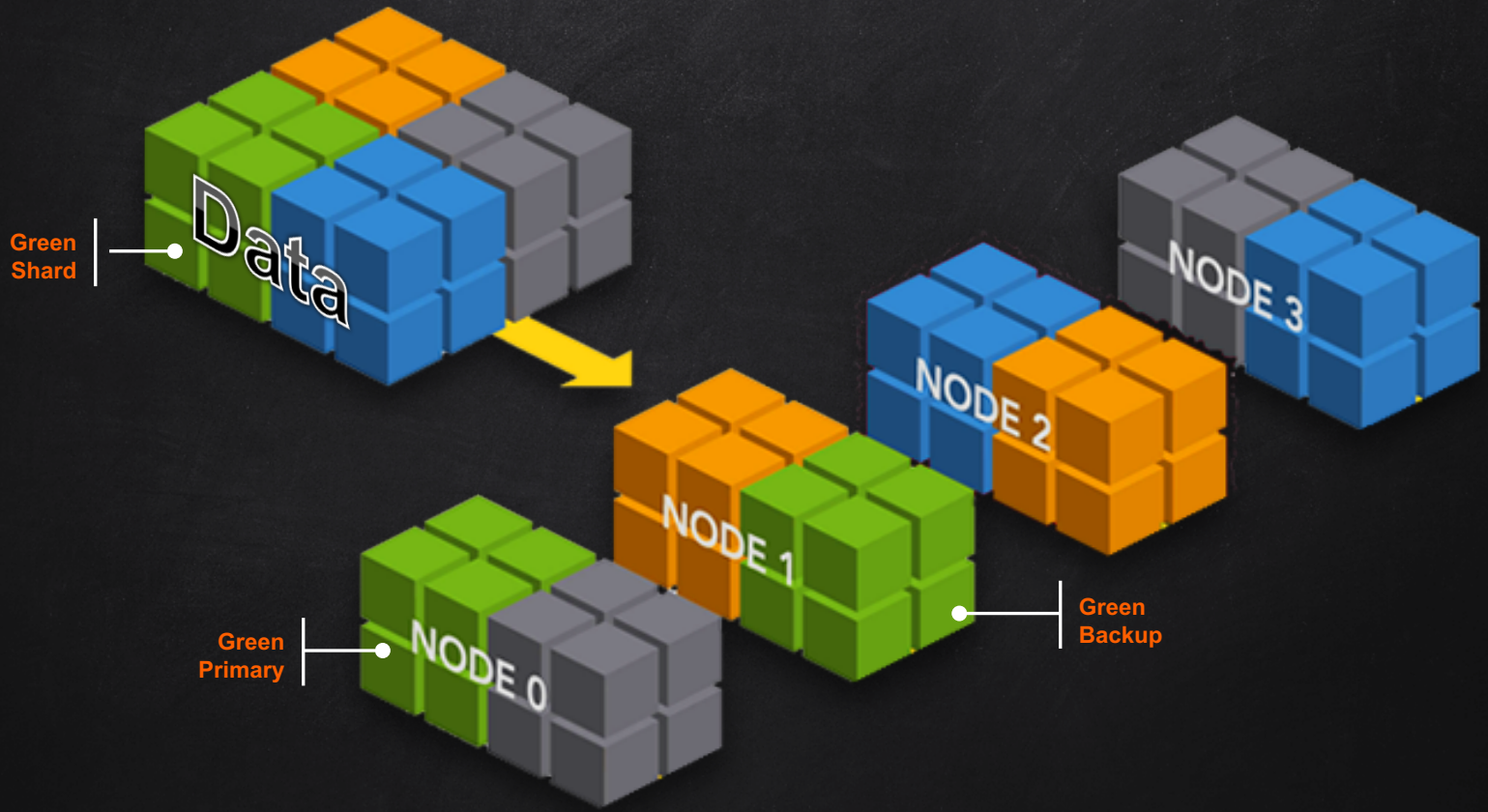
100%

Open Source


Scale-Out Computing


Scale-Up Computing


I DON'T ALWAYS BACKUP THE DATA
BUT WHEN I DO I BACKUP IT IN-MEMORY

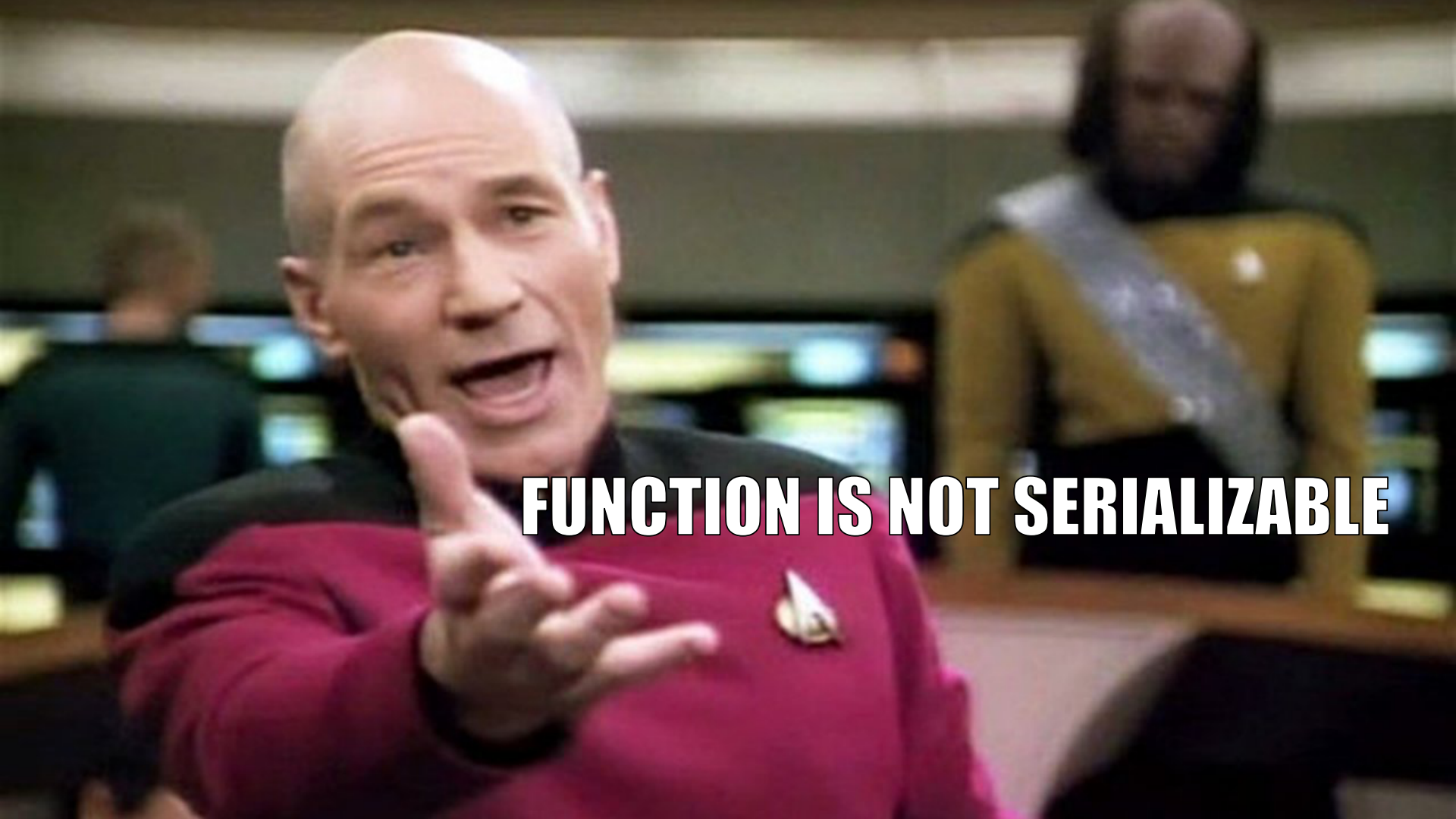

Green
Shard
Data
NODE 3
NODE 2
NODE 1
NODE 0
Green
Primary
Green
Backup


I CAN HAZ DEMO?

What's the problem?

Use `IMap.values().stream()`?
Or `IMap.entrySet().stream()`?


ONE DOES NOT SIMPLY
FIT BIG DATA IN ONE MACHINE


FUNCTION IS NOT SERIALIZABLE

```
import java.util.Iterator;
import java.util.Objects;
import java.util.Optional;
import java.util.Spliterator;
import java.util.Spliterators;
import java.util.concurrent.ConcurrentHashMap;
import java.util.function.BiConsumer;
import java.util.function.BiFunction;
import java.util.function.BinaryOperator;
import java.util.function.Consumer;
import java.util.function.Function;
import java.util.function.IntFunction;
import java.util.function.Predicate;
import java.util.function.Supplier;
import java.util.function.ToDoubleFunction;
import java.util.function.ToIntFunction;
import java.util.function.ToLongFunction;
import java.util.function.UnaryOperator;

/** ... */
public interface Stream<T> extends BaseStream<T, Stream<T>> {

    /** ... */
    Stream<T> filter(Predicate<? super T> predicate);

    /** ... */
    <R> Stream<R> map(Function<? super T, ? extends R> mapper);
```

Function is not Serializable

EASY (ACTUALLY, NOT)!

Implement serializable version of the interfaces

Introducing `DistributedStream`

```
public interface DistributedStream<T> extends Stream<T> {

    /** ... */
    DistributedStream<T> filter(Distributed.Predicate<? super T> predicate);

    /** ... */
    <R> DistributedStream<R> map(Distributed.Function<? super T, ? extends R> mapper);

    /** ... */
    DistributedIntStream mapToInt(Distributed.ToIntFunction<? super T> mapper);

    /** ... */
    DistributedLongStream mapToLong(Distributed.ToLongFunction<? super T> mapper);

    /** ... */
    DistributedDoubleStream mapToDouble(Distributed.ToDoubleFunction<? super T> mapper);

    /** ... */
    <R> DistributedStream<R> flatMap(Distributed.Function<? super T, ? extends Stream<? extends
     R>> mapper);

    /** ... */
    DistributedIntStream flatMapToInt(Distributed.Function<? super T, ? extends IntStream> mapper
```


I CAN HAZ DEMO?

Jet Streams

hazelcast JET

JET.HAZELCAST.ORG

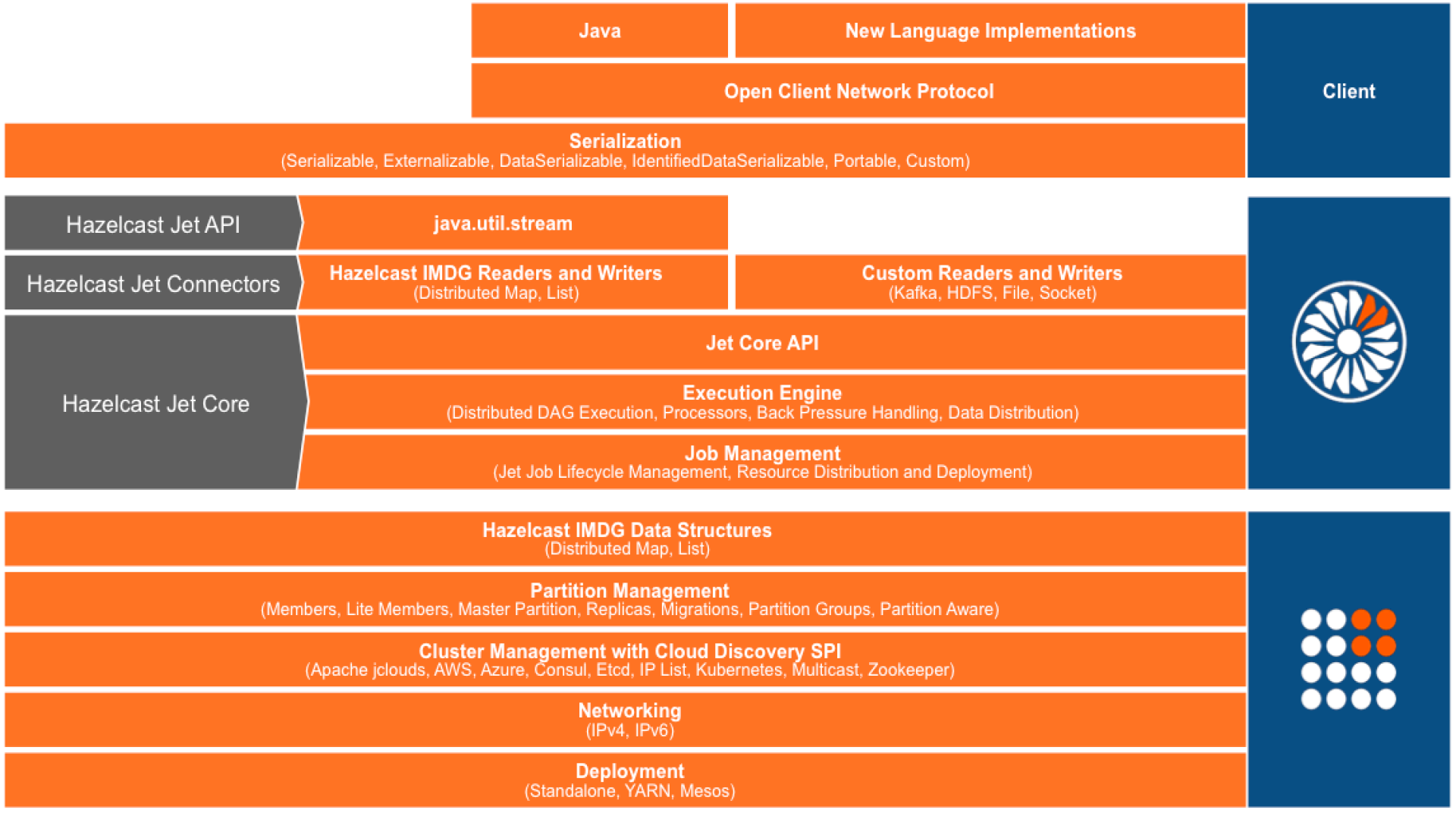
What's Hazelcast Jet?

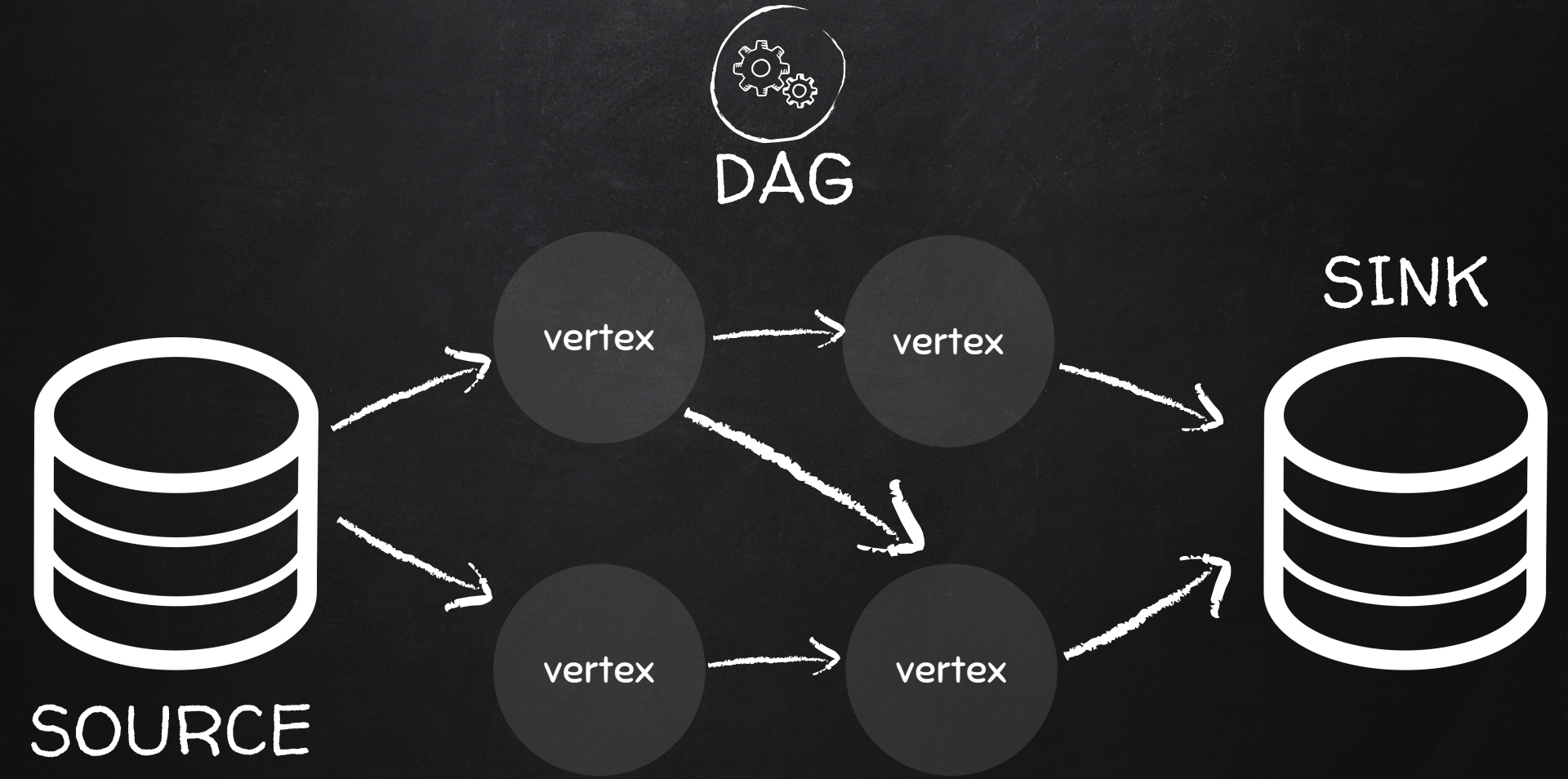
General purpose distributed data processing framework

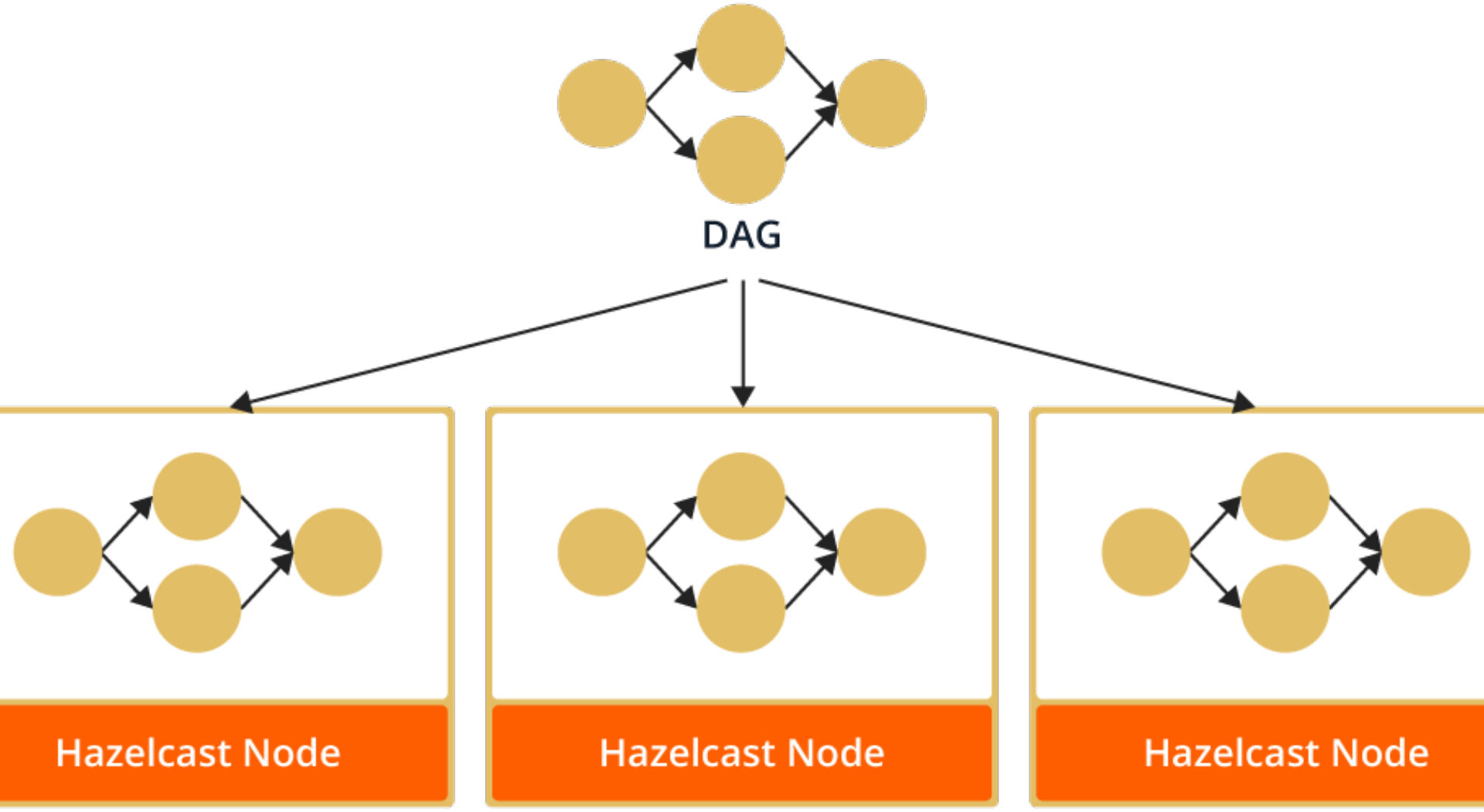
Based on Direct Acyclic Graph to model data flow

Built on top of Hazelcast IMDG

Comparable to Apache Spark or Apache Flink

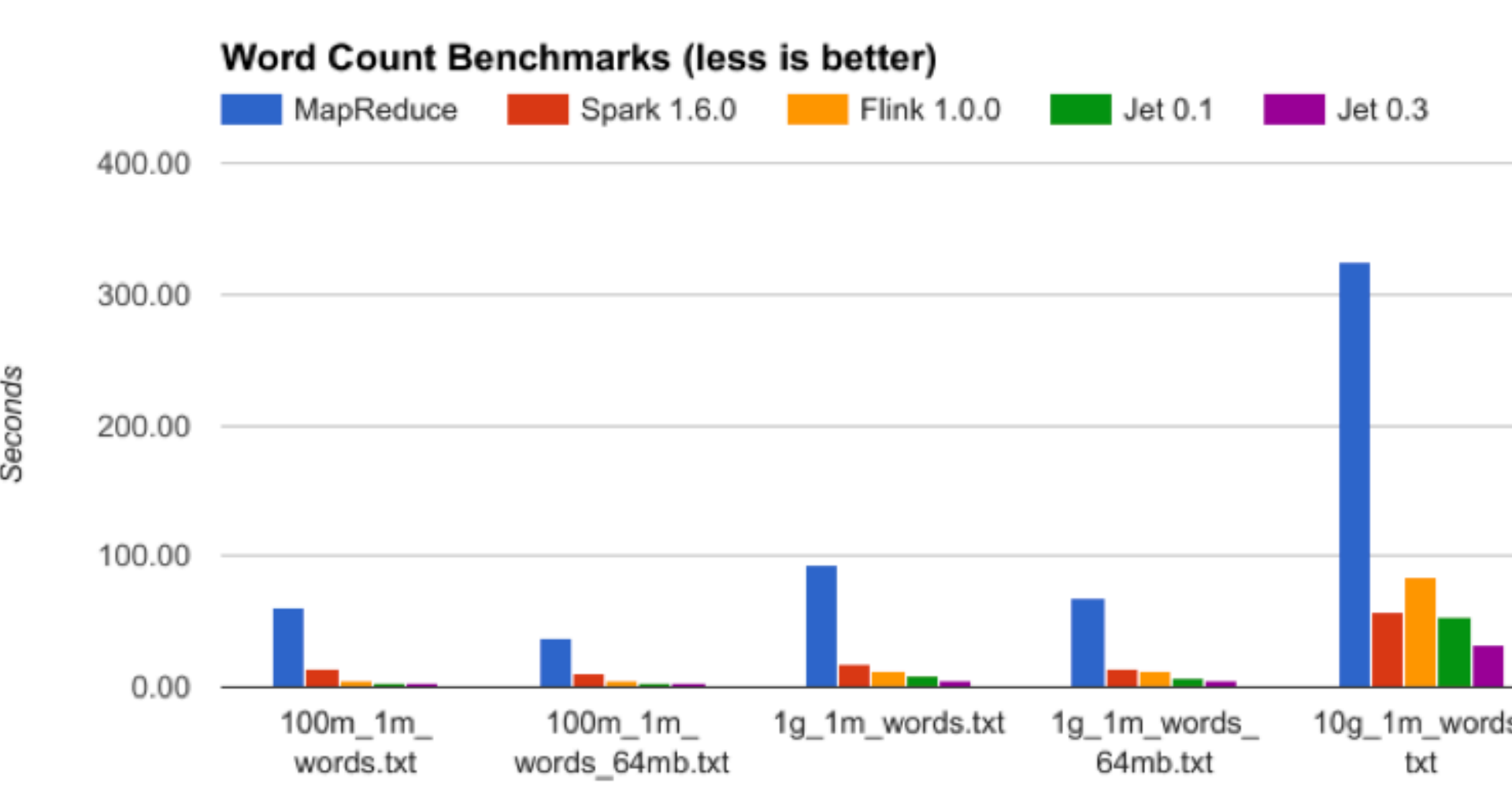

Java
New Language Implementations
Open Client Network Protocol
Client
Serialization
(Serializable, Externalizable, DataSerializable, IdentifiedDataSerializable, Portable, Custom)
Hazelcast Jet API
java.util.stream
Hazelcast Jet Connectors
Hazelcast IMDG Readers and Writers
(Distributed Map, List)
Custom Readers and Writers
(Kafka, HDFS, File, Socket)
Hazelcast Jet Core
Jet Core API
Execution Engine
(Distributed DAG Execution, Processors, Back Pressure Handling, Data Distribution)
Job Management
(Jet Job Lifecycle Management, Resource Distribution and Deployment)
Hazelcast IMDG Data Structures
(Distributed Map, List)
Partition Management
(Members, Lite Members, Master Partition, Replicas, Migrations, Partition Groups, Partition Aware)
Cluster Management with Cloud Discovery SPI
(Apache jclouds, AWS, Azure, Consul, Etcd, IP List, Kubernetes, Multicast, Zookeeper)
Networking
(IPv4, IPv6)
Deployment
(Standalone, YARN, Mesos)


DAG
SINK
vertex
vertex
vertex
vertex
SOURCE


DAG
Hazelcast Node
Hazelcast Node
Hazelcast Node

Benchmarks

Compared to Spark, Flink, Hadoop doing word count, running on a cluster of 9 nodes, 40 cores each


Word Count Benchmarks (less is better)
MapReduce
Spark 1.6.0
Flink 1.0.0
Jet 0.1
Jet 0.3
400.00
300.00
200.00
100.00
0.00
Seconds
100m_1m_words.txt
100m_1m_words_64mb.txt
1g_1m_words.txt
1g_1m_words_64mb.txt
10g_1m_words.txt

Future (It's bright!)

Processing guarantees for stream processing
Streaming features (windowing, triggering)
Higher level streaming and batching APIs
Integration with additional Hazelcast structures (ICache, IQueue ..)

Future (It's bright!)

Event sourcing / CQRS
Off-heap memory support
RxJava
More connectors to additional sources (JMS, JDBC..)

Grab while it's hot!

documentation

jet.hazelcast.org

Source on Github

hazelcast/hazelcast-jet

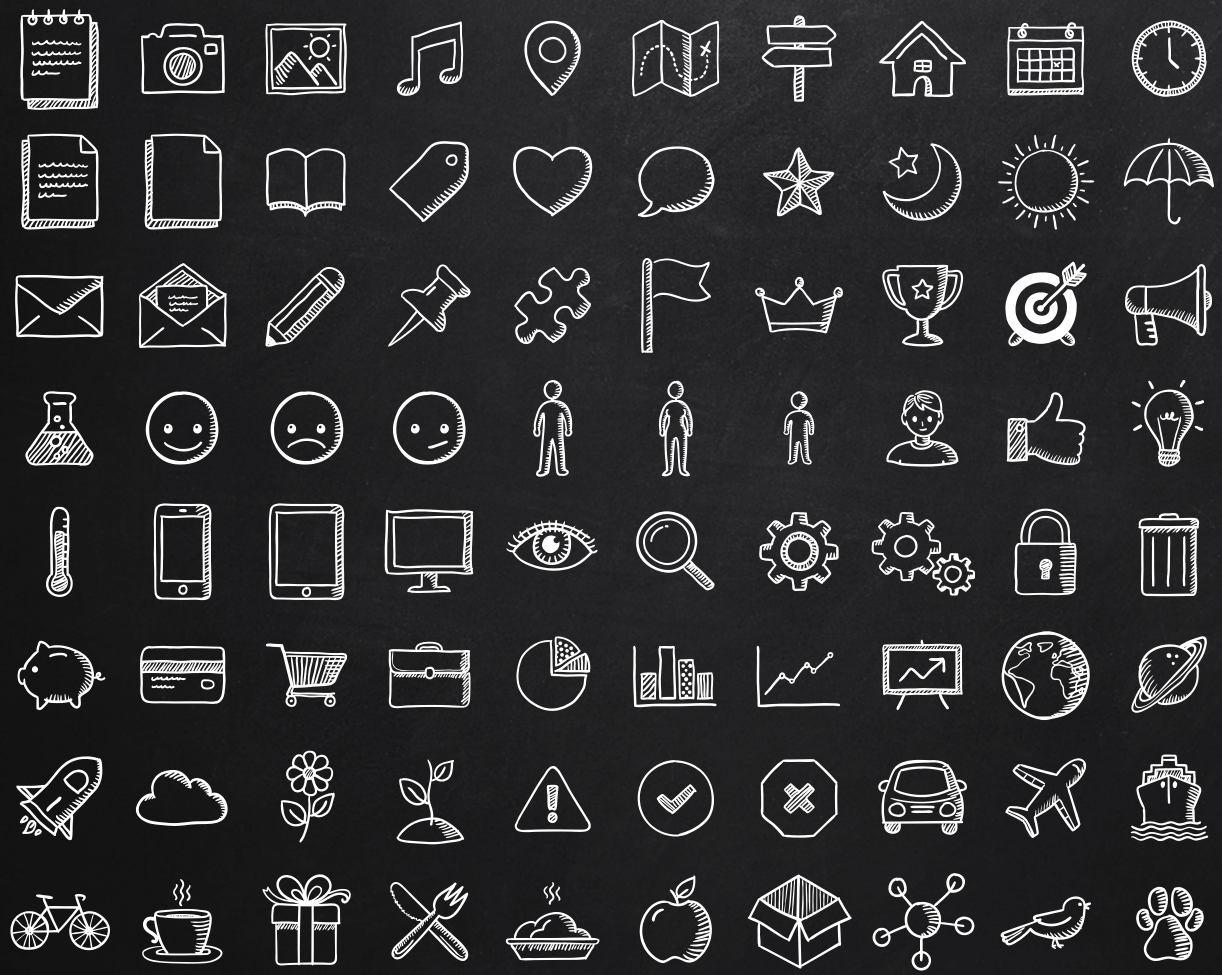
Presentation materials

http://bit.ly/streams_jfokus2017

Conclusion

Java Stream API provides very white range of data processing tools
War And Piece – is a Big (a lot of data) Book!
Now we're pretty sure that Andrew and Pierre are the main characters


HTTP://BIT.LY/STREAMS_JFOKUS2017
@gAmUssA @hazelcast #jfokus #hazelcastjet



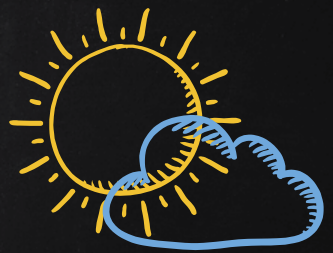
SlidesCarnival icons are editable shapes.

This means that you can:

- Resize them without losing quality.
- Change fill color and opacity.

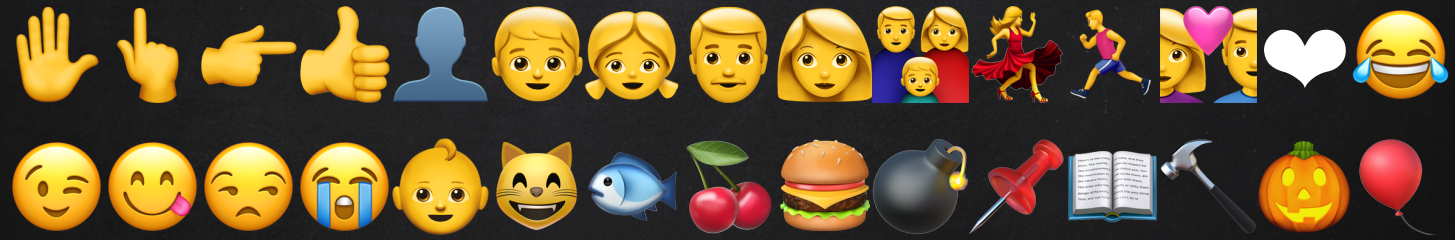
Isn't that nice? :)

Examples:



Now you can use any emoji as an icon!
And of course it resizes without losing quality and you can change the color.

How? Follow Google instructions
https://twitter.com/googledocs/status/730087240156643328



and many more...

Extra graphics