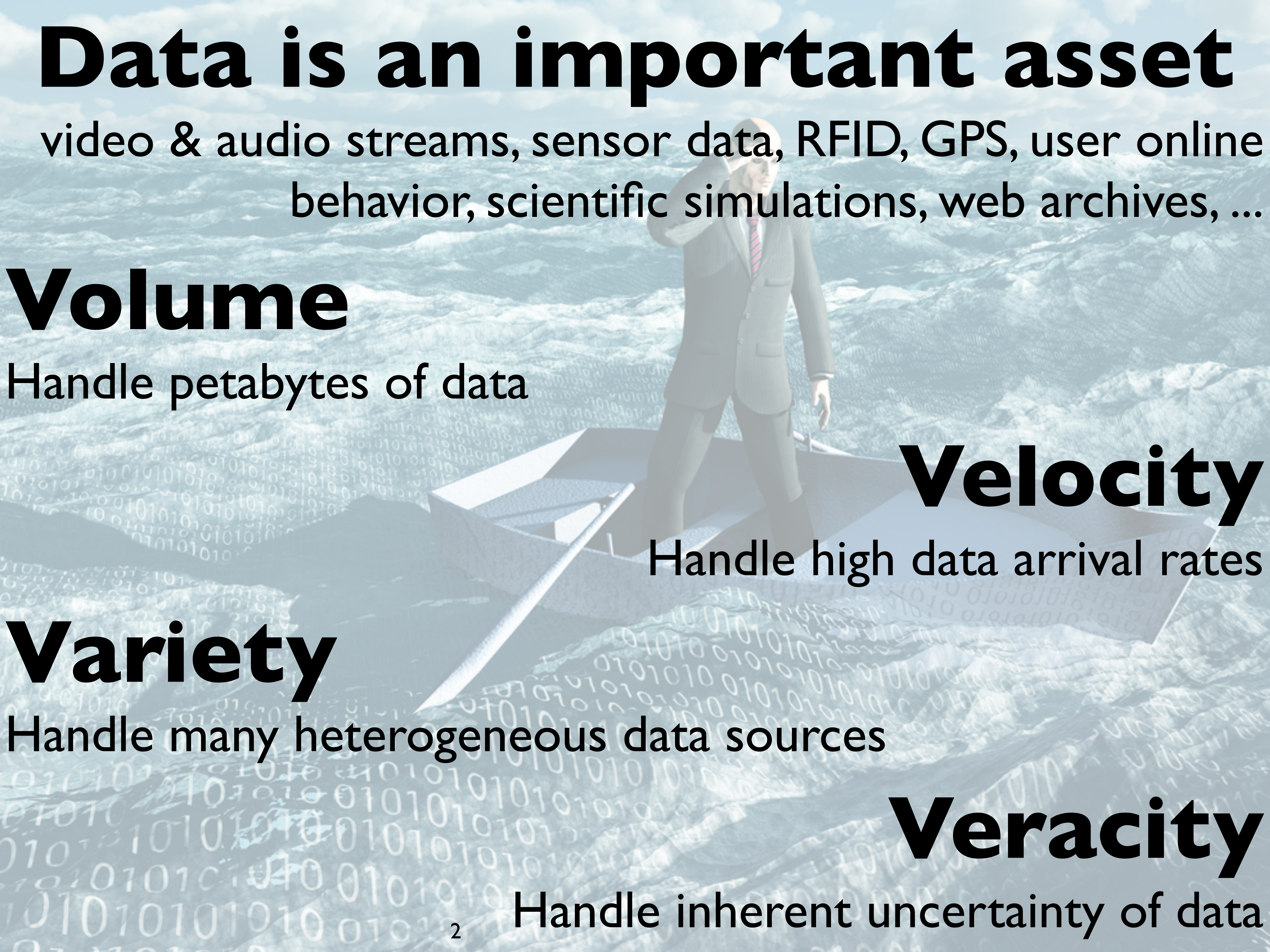
Ask “what,” not “how”

Kostas Tzoumas



Data is an important asset

video & audio streams, sensor data, RFID, GPS, user online behavior, scientific simulations, web archives, ...

Volume

Handle petabytes of data

Velocity

Handle high data arrival rates

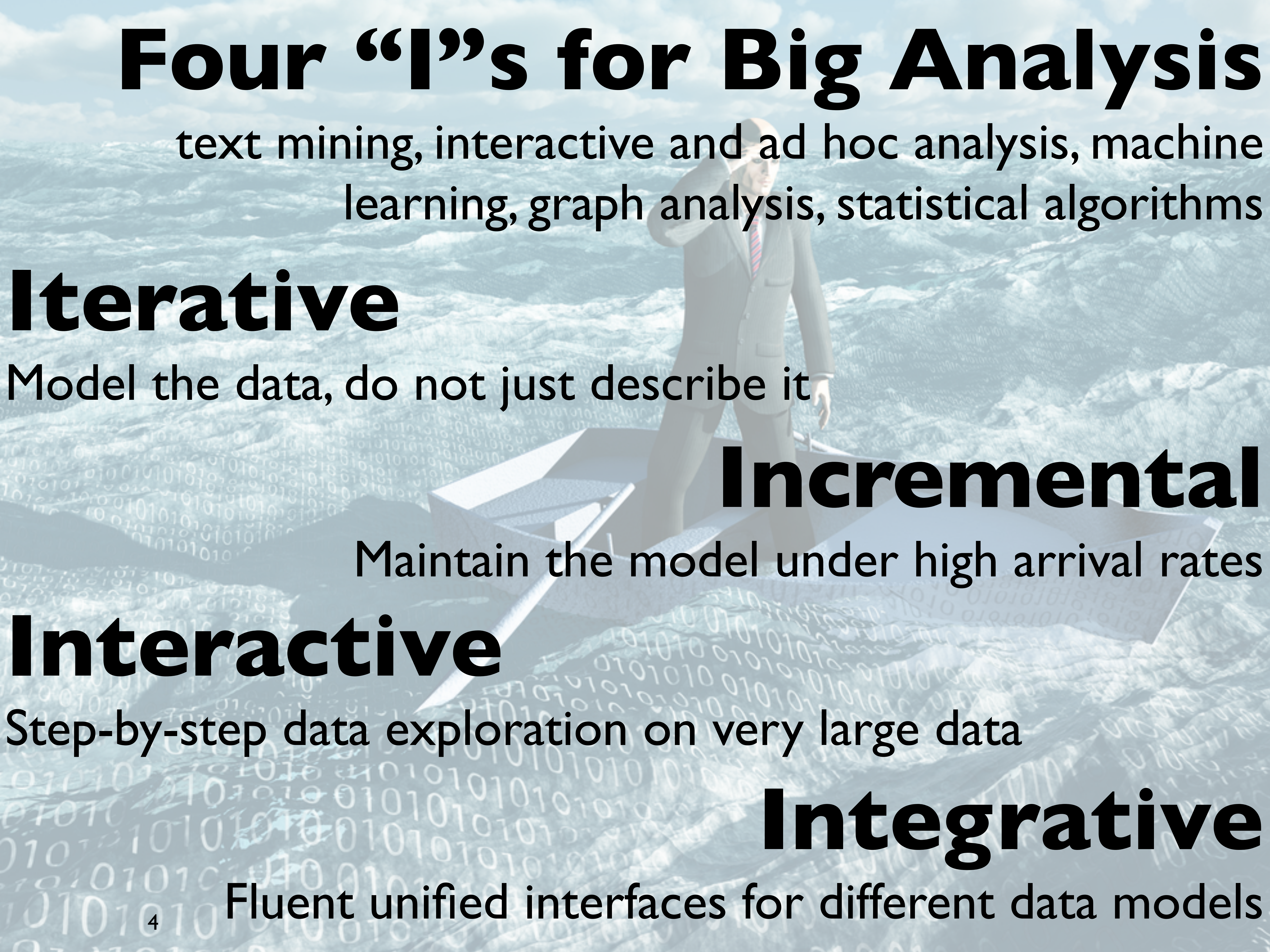
Variety

Handle many heterogeneous data sources

Veracity

Handle inherent uncertainty of data

Data Analysis



Four “I”s for Big Analysis

text mining, interactive and ad hoc analysis, machine learning, graph analysis, statistical algorithms

Iterative

Model the data, do not just describe it

Incremental

Maintain the model under high arrival rates

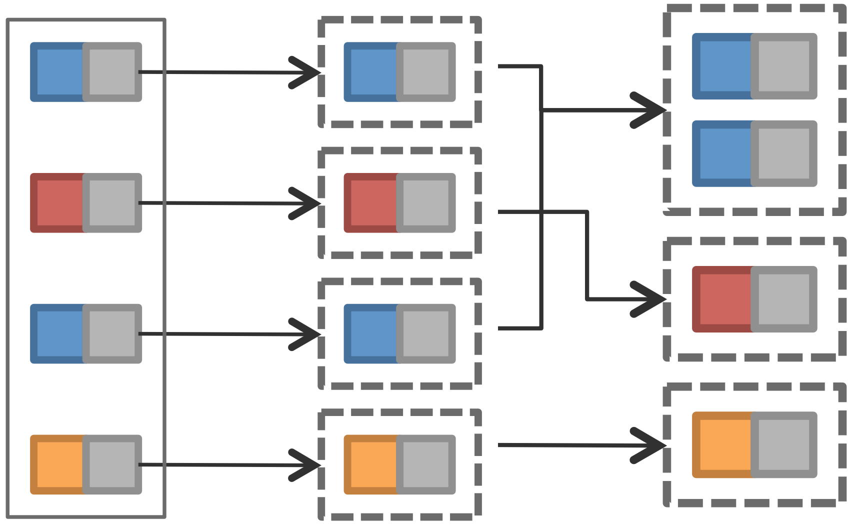
Interactive

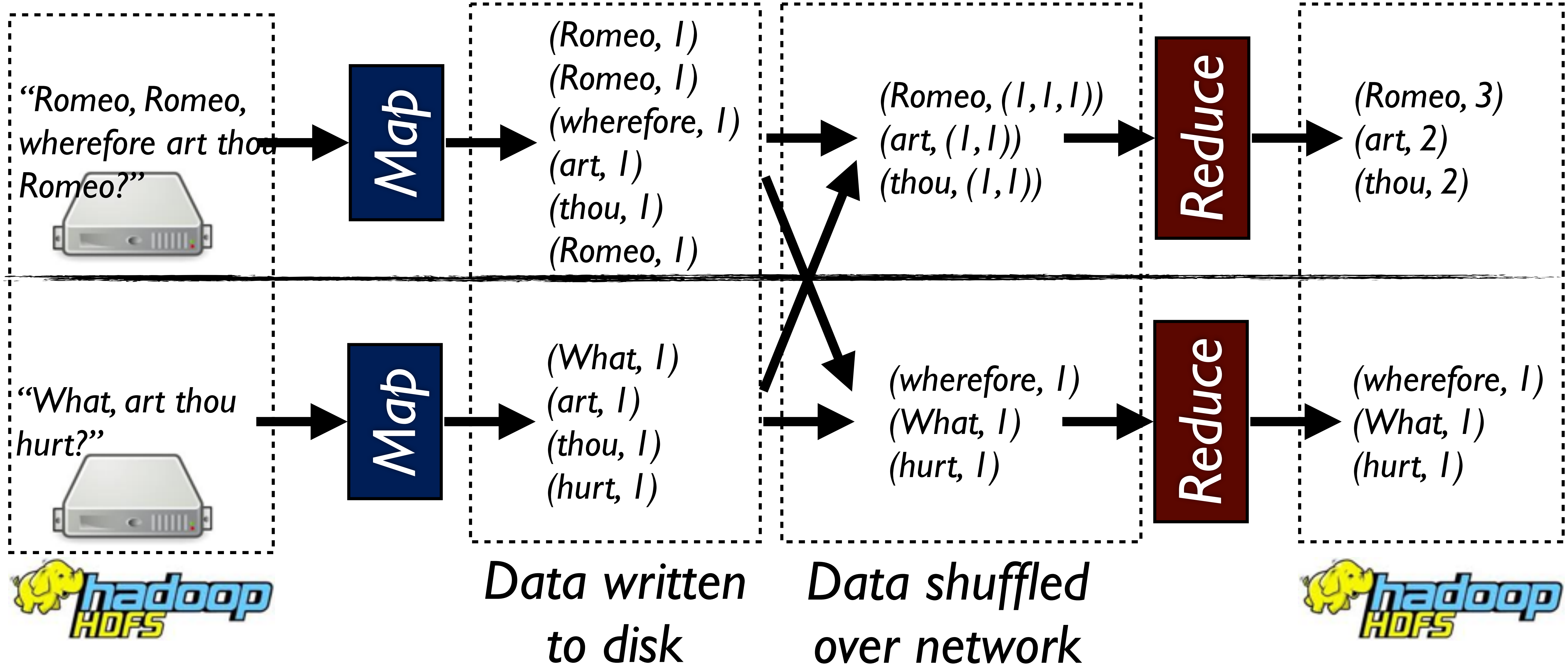
Step-by-step data exploration on very large data

Integrative

Fluent unified interfaces for different data models

MapReduce and Hadoop





"Romeo, Romeo, wherefore art thou Romeo?"

Map

(Romeo, 1)
(Romeo, 1)
(wherefore, 1)
(art, 1)
(thou, 1)
(Romeo, 1)

(Romeo, (1,1,1))
(art, (1,1))
(thou, (1,1))

Reduce

(Romeo, 3)
(art, 2)
(thou, 2)

"What, art thou hurt?"

Map

(What, 1)
(art, 1)
(thou, 1)
(hurt, 1)

(wherefore, 1)
(What, 1)
(hurt, 1)

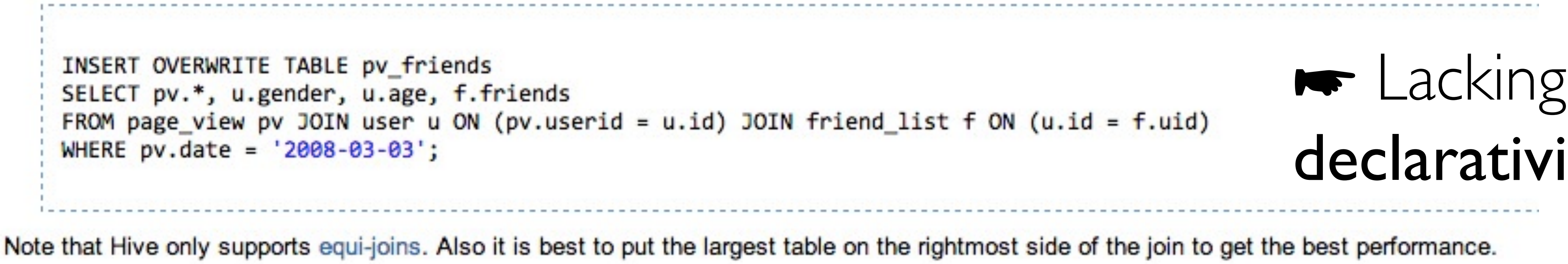
Reduce

(wherefore, 1)
(What, 1)
(hurt, 1)

Data written to disk

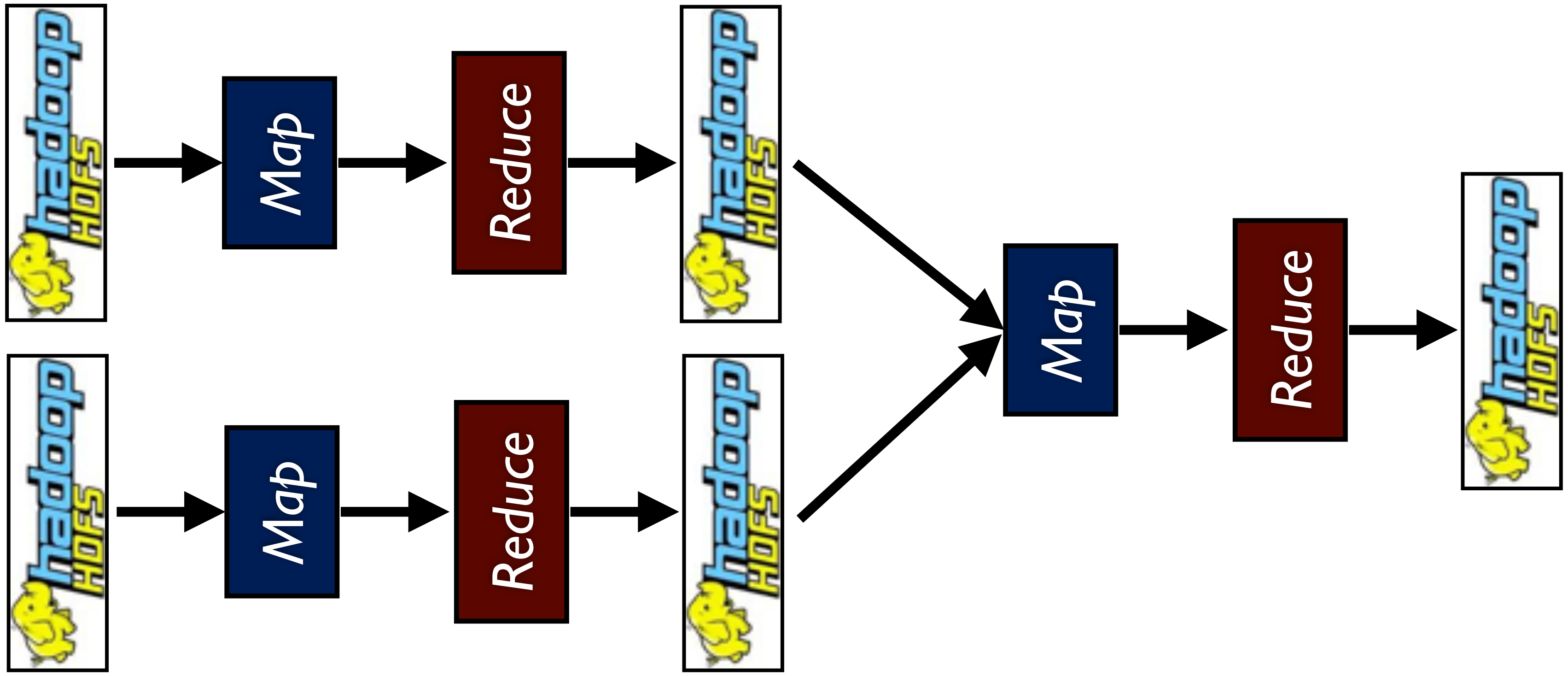
Data shuffled over network

SQL analytics with Hadoop



```
INSERT OVERWRITE TABLE pv_friends
SELECT pv.*, u.gender, u.age, f.friends
FROM page_view pv JOIN user u ON (pv.userid = u.id) JOIN friend_list f ON (u.id = f.uid)
WHERE pv.date = '2008-03-03';
```

Note that Hive only supports equi-joins. Also it is best to put the largest table on the rightmost side of the join to get the best performance.



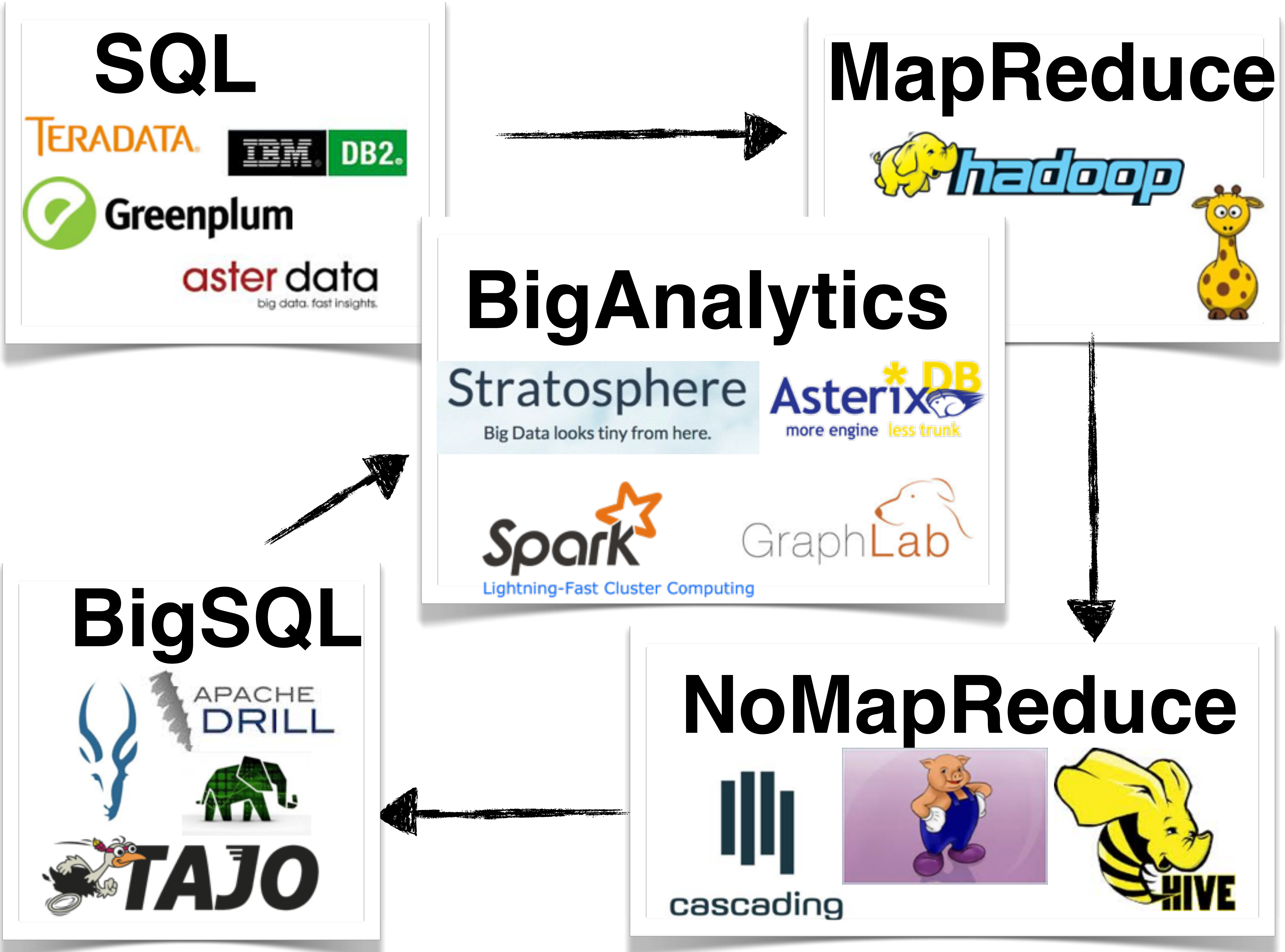
Pitfalls:

☛ Lacking in **declarativity**

☛ HDFS-based data exchange

☛ Sort the only grouping operator

☛ Hadoop engine tailored to simple aggregations



SQL

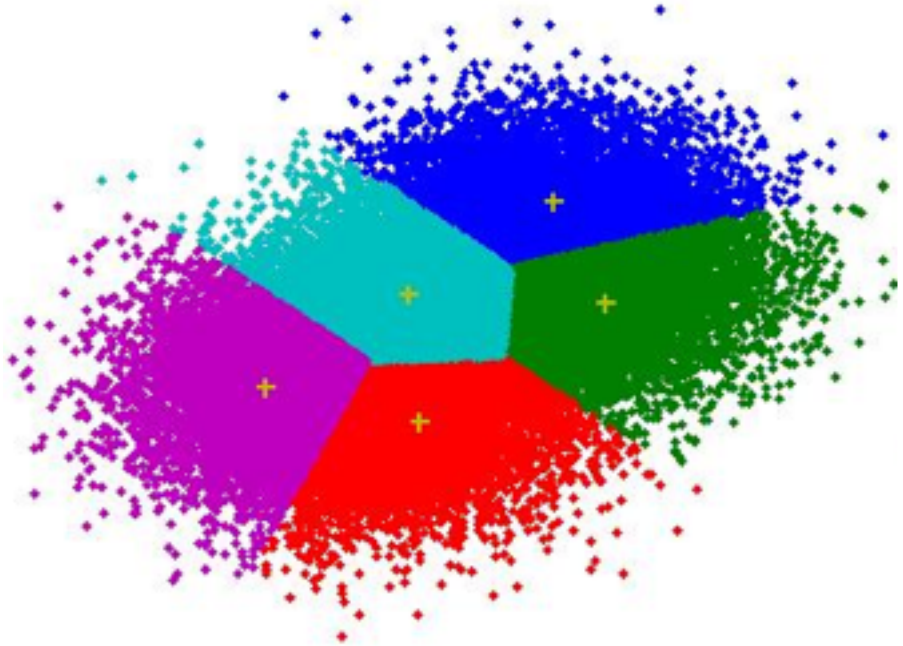
MapReduce

BigAnalytics

NoMapReduce

BigSQL

Advanced Analytics



Analytics that **model the data** to reveal hidden relationships**, not just describe** the data.

E.g., machine learning, statistics, graph analysis

Increasingly important from a market perspective.

Very different than SQL analytics: different languages and access patterns (iterative vs. one-pass programs).

Hadoop toolchain poor; R, Matlab, etc not parallel.

Use case in all verticals

Manufacturing

Example: Data-driven quality control and assurance, demand forecasting, sales and operation planning, process optimization

Retail

Example: Improve campaign ROI by optimizing advertising channels, market basket analysis, fraud detection, social trend analysis, product recommendation

Media and Communications

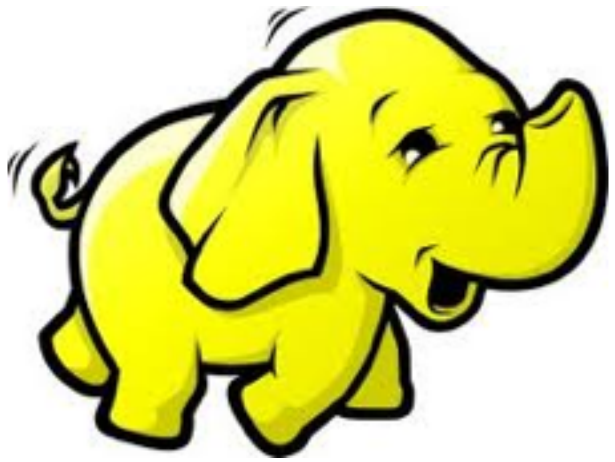
Example: Risk management, analytics on phone call logs, risk management, sentiment analysis, clickstream and call analysis

Travel and tourism

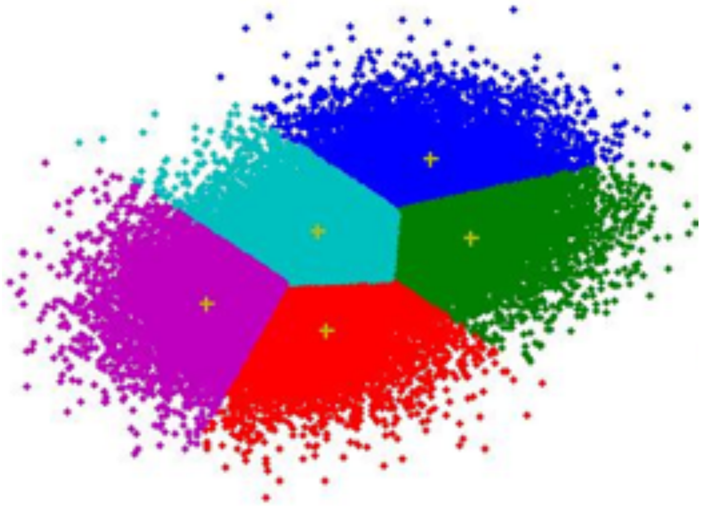
Example: Improve personalized customer experience in hotels, estimate no-show in flights, route planning

Social and e-commerce

Example: Targeted customer experience, explore new business models, real-time recommendations, social graph analysis, game analytics



Big data lives in Hadoop. Hadoop clusters offer very **low effective storage cost**, and are becoming a **data vortex**, attracting **cross-departmental data**.



Companies want to perform **advanced and predictive analytics** to **maximize ROI of their data assets by modeling the data, not just describing it.**

How do we bring advanced analytics to the world of big data?

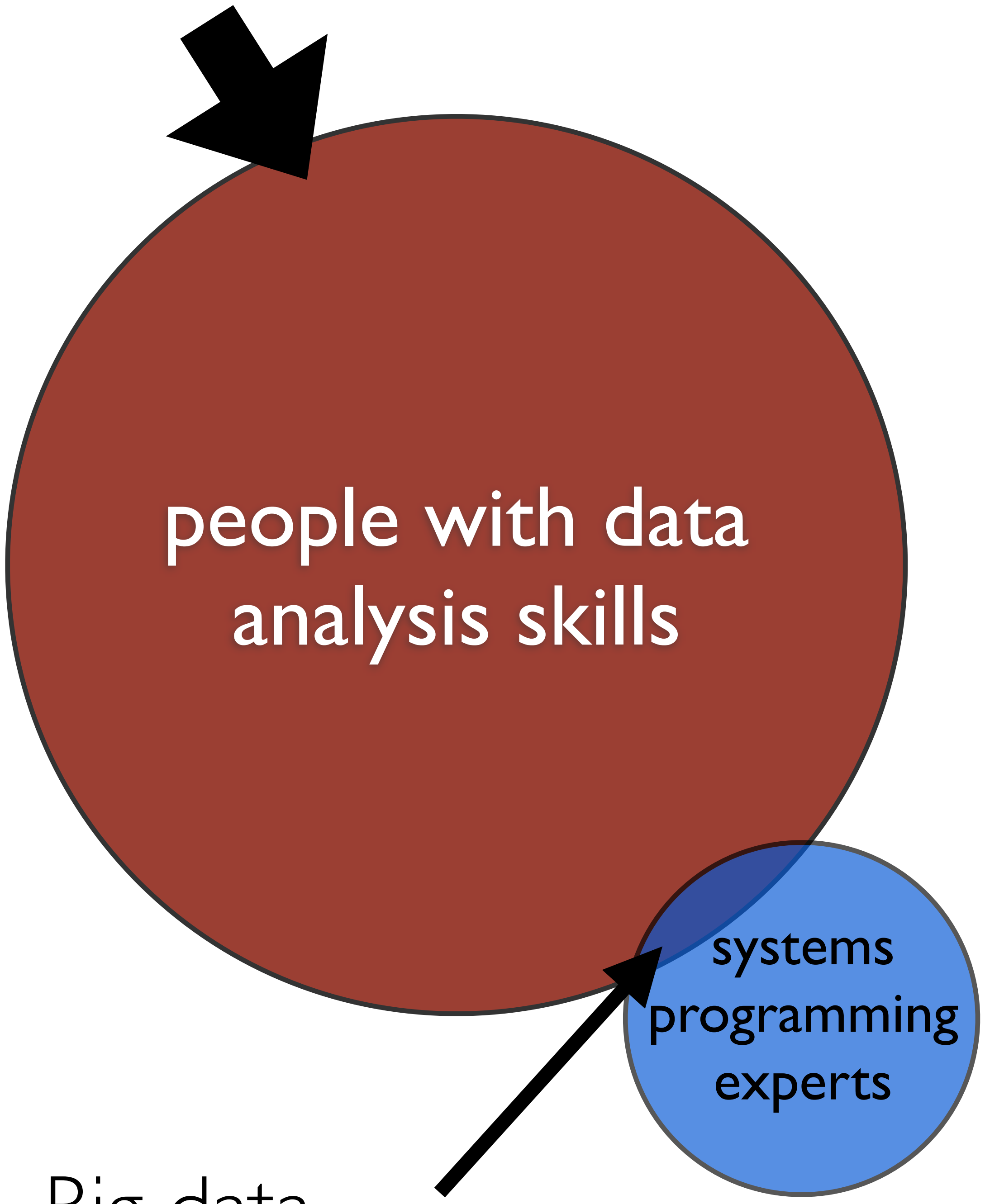
What, not how

Recipe for success: **declarativity**

User specifies **what** information to extract out of the data, **not how** the system extracts the information.

This is what relational databases pioneered in the 70s resulting in a vibrant research community and a billion dollar industry.

Big data consumers in the future

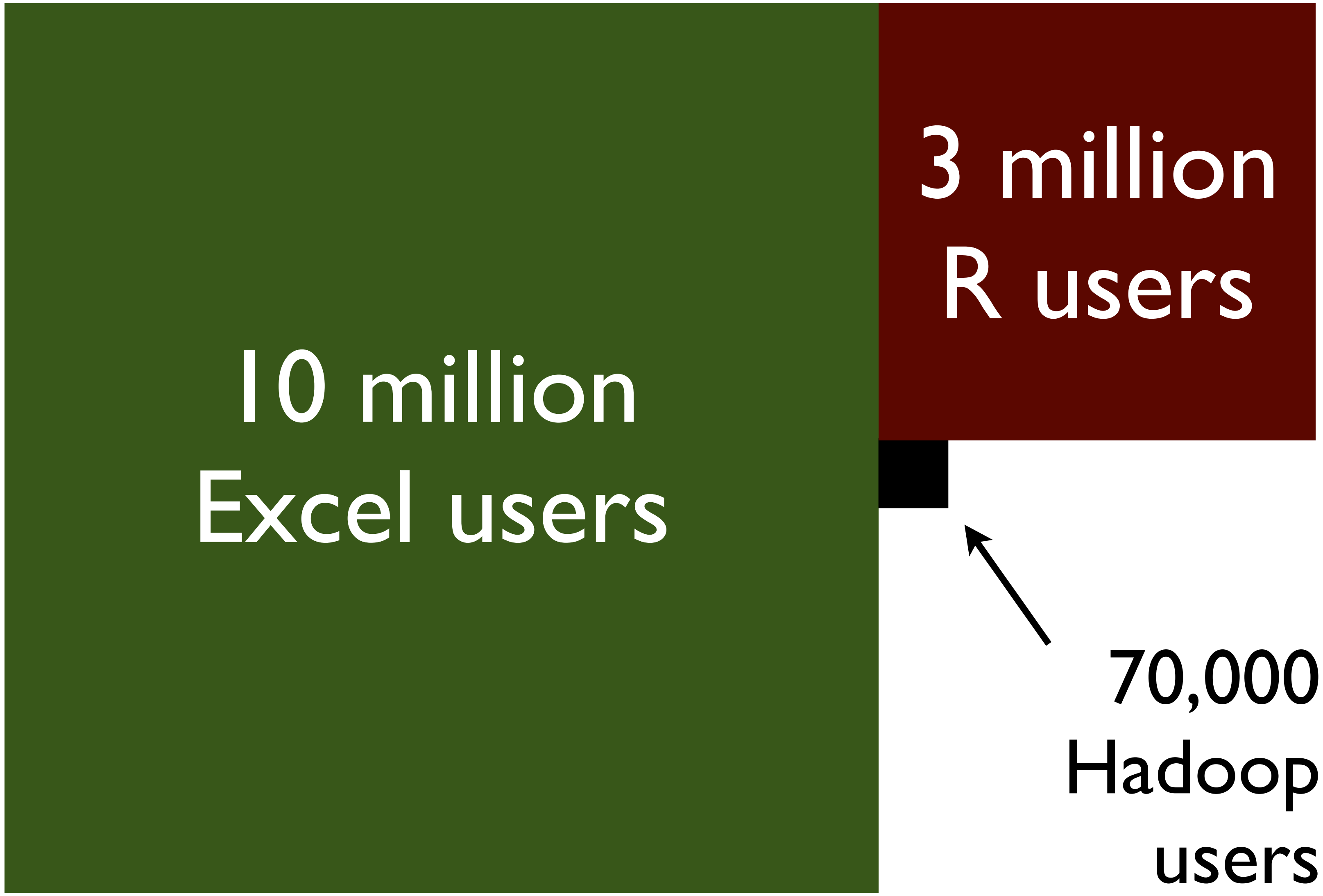


people with data analysis skills

systems programming experts

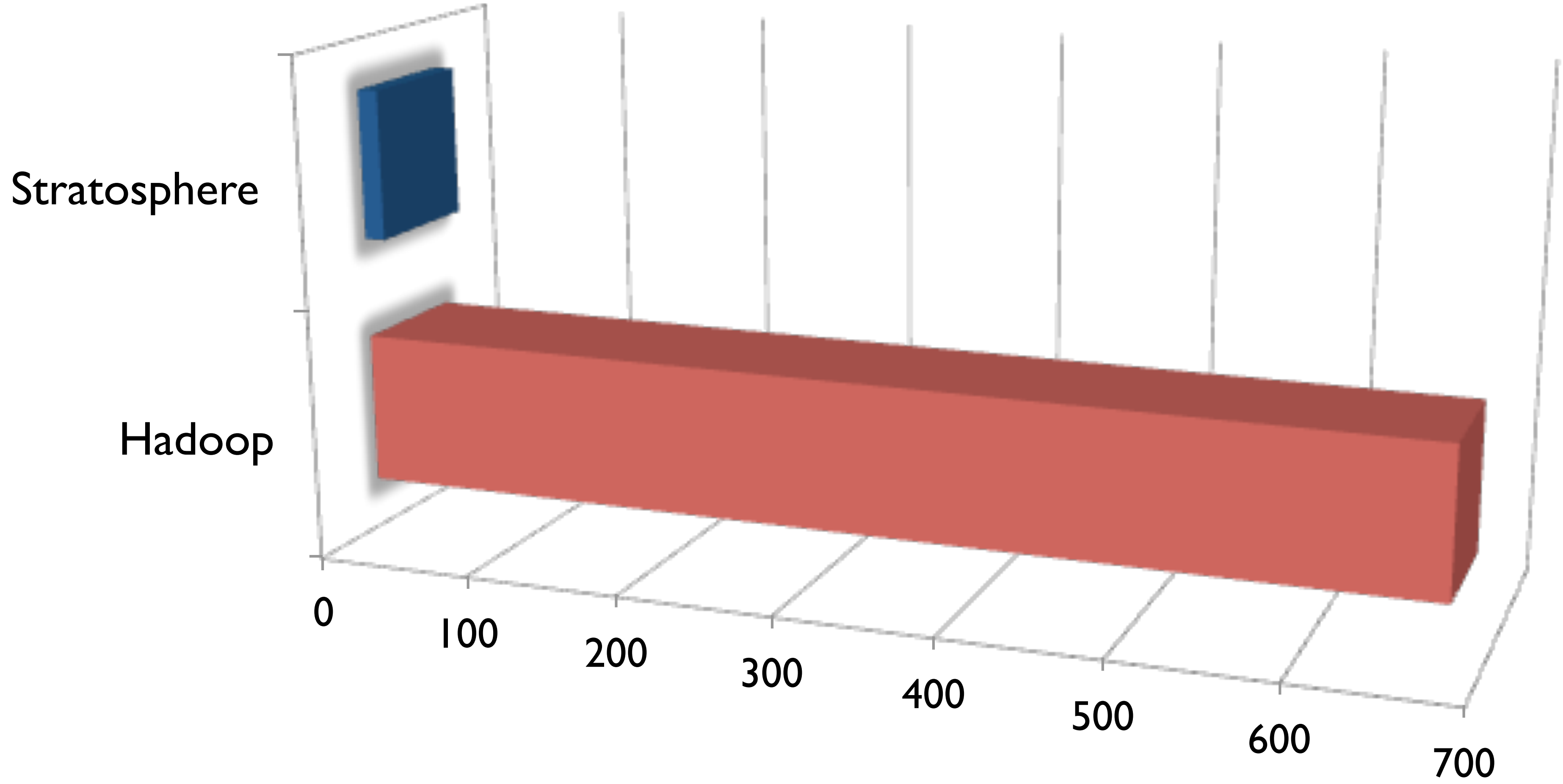
Big data consumers now

Desiderata for next-gen big data platforms: *Usability*



"the market faces certain challenges such as ***unavailability of qualified and experienced work professionals****, who can effectively handle the Hadoop architecture."*

Desiderata for next-gen big data platforms: *Performance*



Performance difference **from days to minutes** enables **real time decision making** and widespread use of data within the organization.

How to lift declarativity from the closed world of relational algebra to the open world of advanced analytics.

Step 1: Specify

```
// get the customers with their debit
val debits: (String, Double) = sql(
    "SELECT customerId, debit FROM customer_accounts;")
// get the number of warned invoices in the last
// 12 and 6 months
val warnings: (String, Int, Int) = sql
    "SELECT R12.customerId, R12.cnt, R6.cnt
            FROM (…) R12 LEFT OUTER JOIN (…) R6
              ON (R6.customerId = R12.customerId);")
// number of contracts a customer has
val numContracts : (String, Int) = sql(
    "SELECT customerId, numContracts FROM customers;")
```

Unify data and programming models in a declarative abstraction.

SQL for extracting enterprise data from databases.

```
// join the data into one data point
case class DataPoint(x: Vector, y: Double)

val dataPoints = numContracts
  join warnings
  where {_._1} isEqualTo {_._1}
  join debits
  where {_._1} isEqualTo {_._1}
  map { (x,y,z) => DataPoint(Vector(x._2, y._2, y._3),
                             if (z._2 > X) 1 else 0) }
```

General-purpose programming for feature extraction and normalization.

```
// run regression with dimensionality 3 for 40 iterations
val weights: Vector = LogRegression(3, dataPoints, 40)
```

Statistical libraries for advanced analysis.

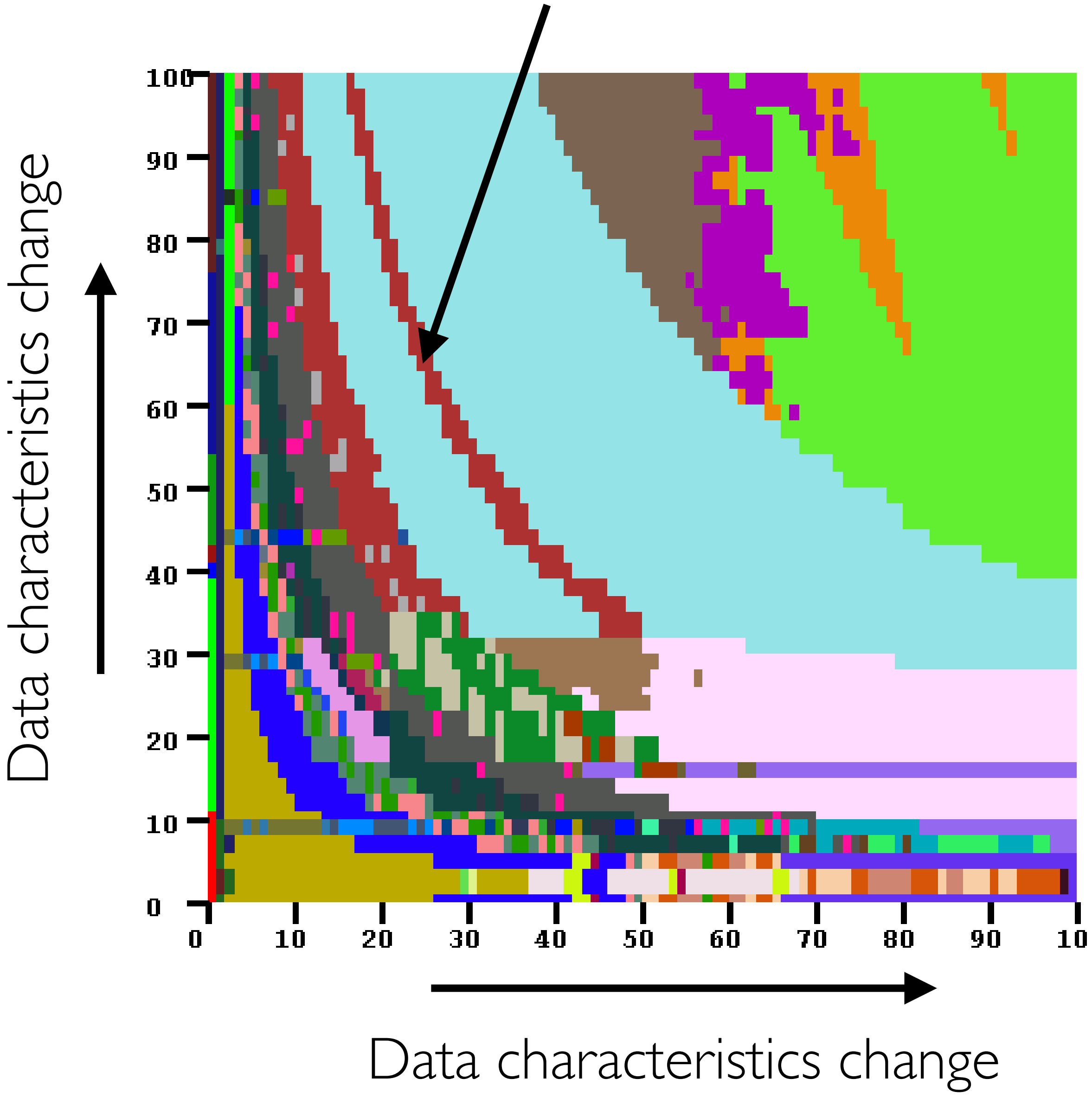
First step for declarative analytics

Scala: functional and object-oriented JVM language, excellent basis for domain-specific language development. Coolest kid in the block ☺

Feels like a scripting language, but is not restricted to a fixed data model like Pig, Hive, etc.

Scala's extensible compiler architecture is a good match for implementing optimizers.

Step 2: Optimize



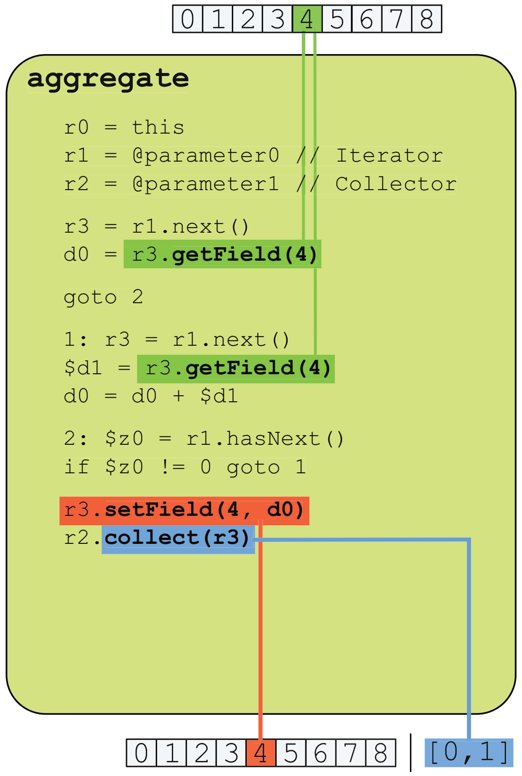
Each color is a differently written program that produces the same result but has very different performance depending on small changes in the data set and the analysis requirements

Query optimizers: the enabling technology for SQL data warehousing and BI

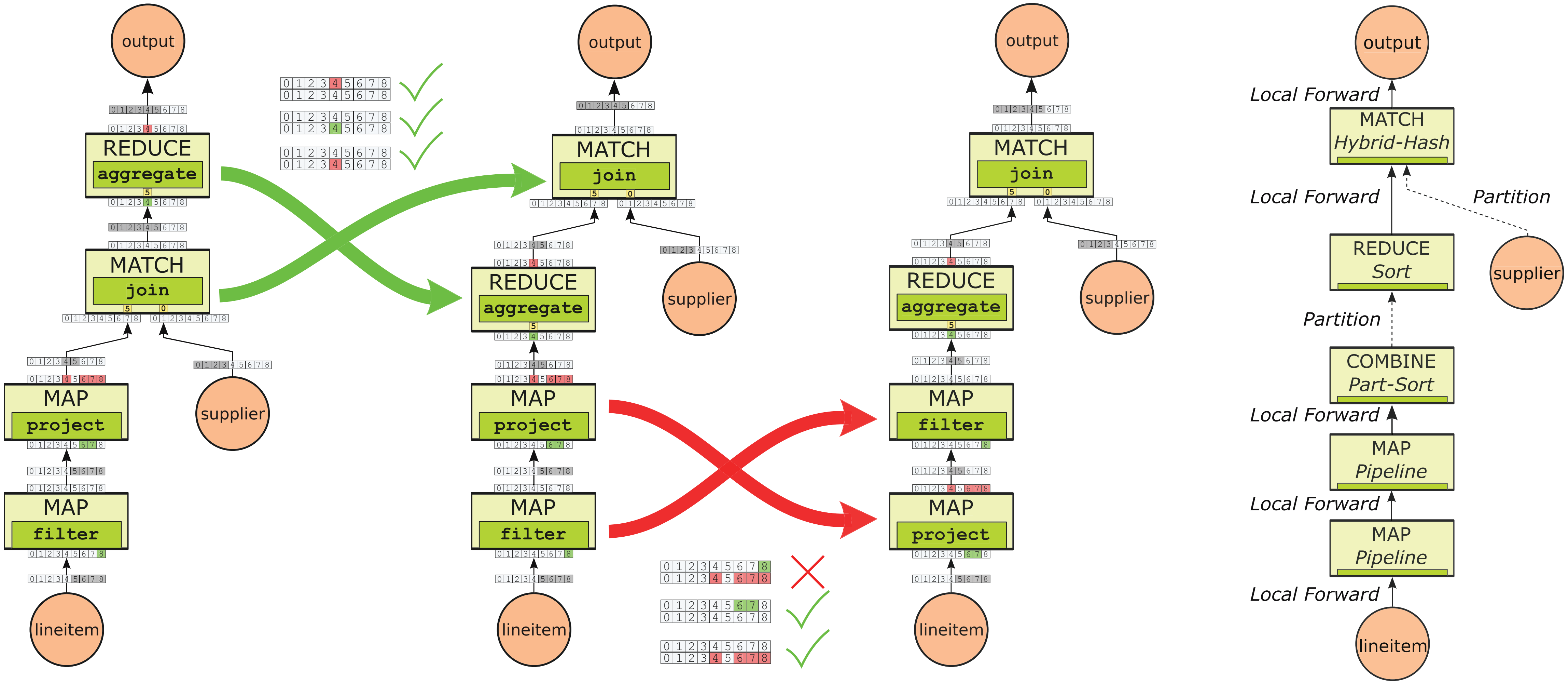
Successful industrial application of artificial intelligence

Currently, no other system can optimize non-relational data analysis programs.

Use a **combination of compiler and database technology** to lift optimization beyond relational algebra. **Derive properties of user-defined functions** via code analysis and use these to **mimic a relational database optimizer.**



```
aggregate

  r0 = this
  r1 = @parameter0 // Iterator
  r2 = @parameter1 // Collector

  r3 = r1.next()
  d0 = r3.getField(4)

  goto 2

  1: r3 = r1.next()
  $d1 = r3.getField(4)
  d0 = d0 + $d1

  2: $z0 = r1.hasNext()
  if $z0 != 0 goto 1

  r3.setField(4, d0)
  r2.collect(r3)
```

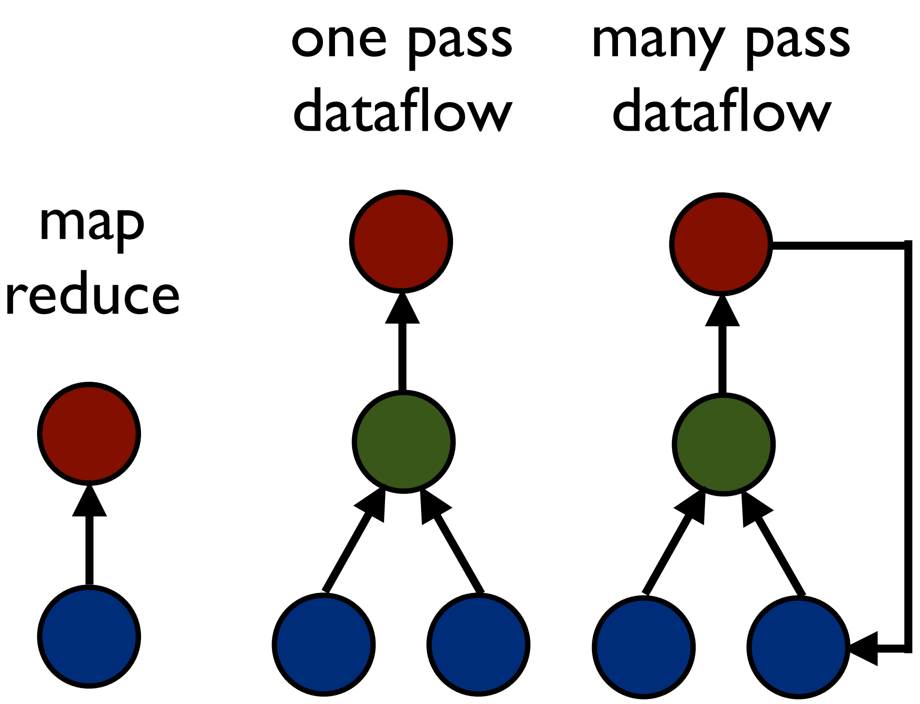


Step 3: Execute

A fast, massively parallel database-inspired backend.

Truly **scales to disk-resident large data sets.**

Built-in support for **iterative programs:** predictive and advanced analytics (machine learning, graph processing, stats) are all iterative.



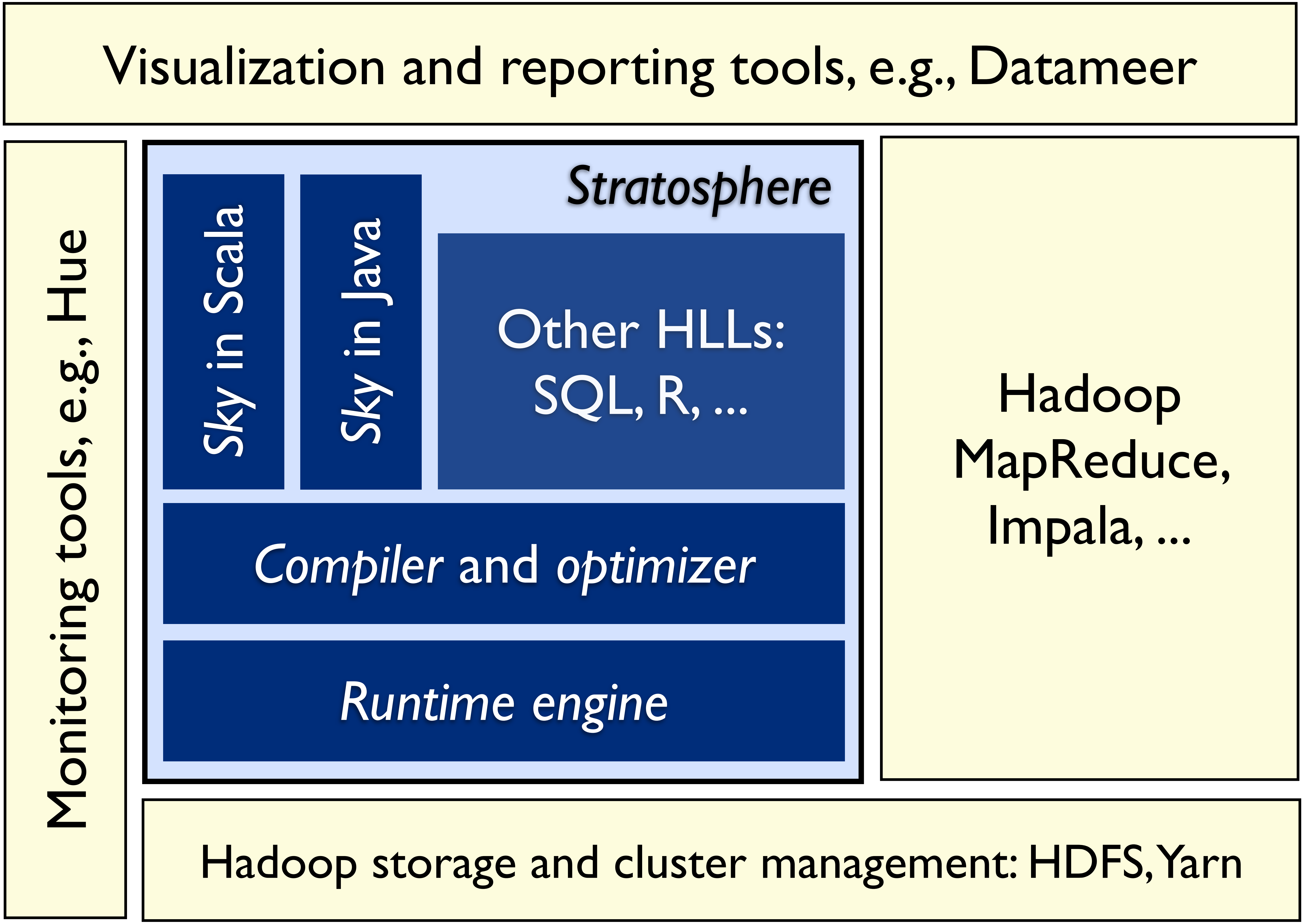
	MapReduce	Impala, ...	Stratosphere
Text	✔	✔	✔
Aggregation	✔	✔	✔
ETL	✔	✔	✔
SQL	Hive is too slow	✔	✔
Advanced analytics	Mahout is slow and low level	Madlib is too slow	✔

Stratosphere is an **award-winning open-source** platform: **15 man-years of R&D, 150k LOC, 3 million € behind it.**

HP Open Innovation Award

IBM Faculty Award

Stratosphere is **the only Hadoop-compatible next-generation big data analytics platform developed in Europe** that you can **download and use right now.**



Visualization and reporting tools, e.g., Datameer

Monitoring tools, e.g., Hue

Stratosphere

Sky in Scala

Sky in Java

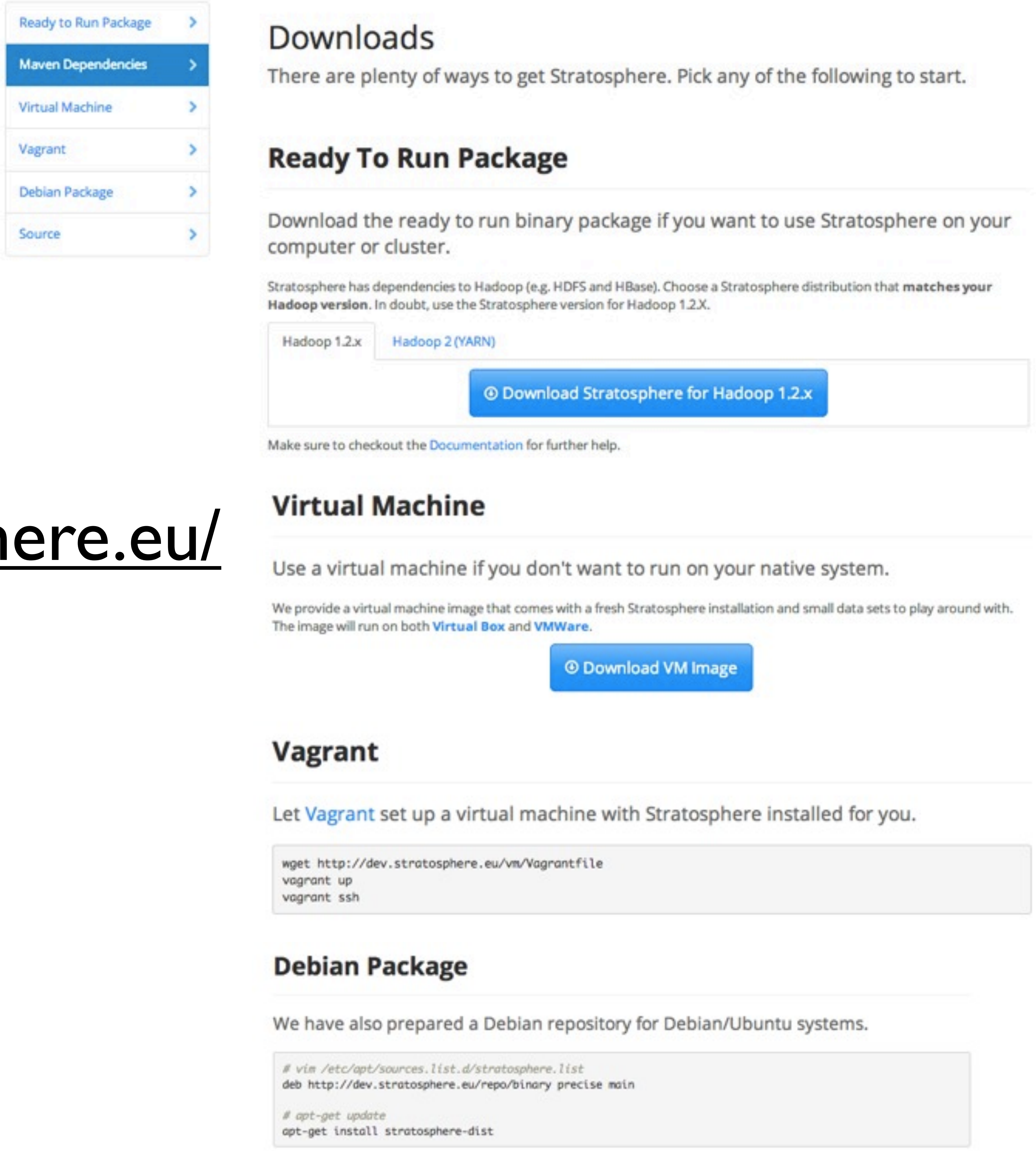
Other HLLs:
SQL, R, ...

Compiler and *optimizer*

Runtime engine

Hadoop
MapReduce,
Impala, ...

Hadoop storage and cluster management: HDFS, Yarn



Downloads

There are plenty of ways to get Stratosphere. Pick any of the following to start.

Ready To Run Package

Download the ready to run binary package if you want to use Stratosphere on your computer or cluster.

Stratosphere has dependencies to Hadoop (e.g. HDFS and HBase). Choose a Stratosphere distribution that **matches your Hadoop version**. In doubt, use the Stratosphere version for Hadoop 1.2.X.

Make sure to checkout the Documentation for further help.

Virtual Machine

Use a virtual machine if you don't want to run on your native system.

We provide a virtual machine image that comes with a fresh Stratosphere installation and small data sets to play around with. The image will run on both **Virtual Box** and **VMWare**.

Vagrant

Let Vagrant set up a virtual machine with Stratosphere installed for you.

```
wget http://dev.stratosphere.eu/vm/Vagrantfile
vagrant up
vagrant ssh
```

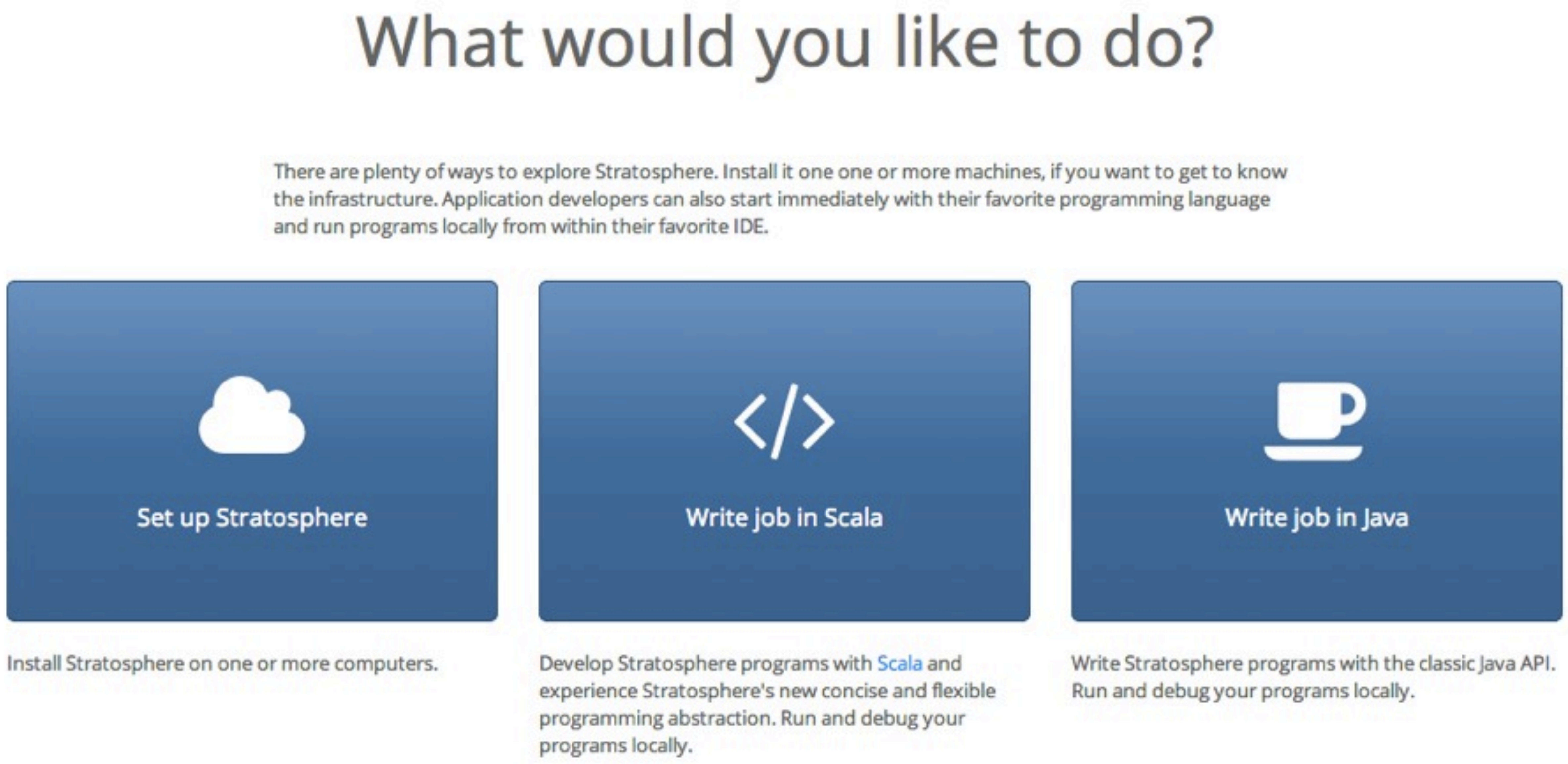
Debian Package

We have also prepared a Debian repository for Debian/Ubuntu systems.

```
# vim /etc/apt/sources.list.d/stratosphere.list
deb http://dev.stratosphere.eu/repo/binary precise main

# apt-get update
apt-get install stratosphere-dist
```

www.stratosphere.eu/downloads

www.stratosphere.eu/quickstart



What would you like to do?

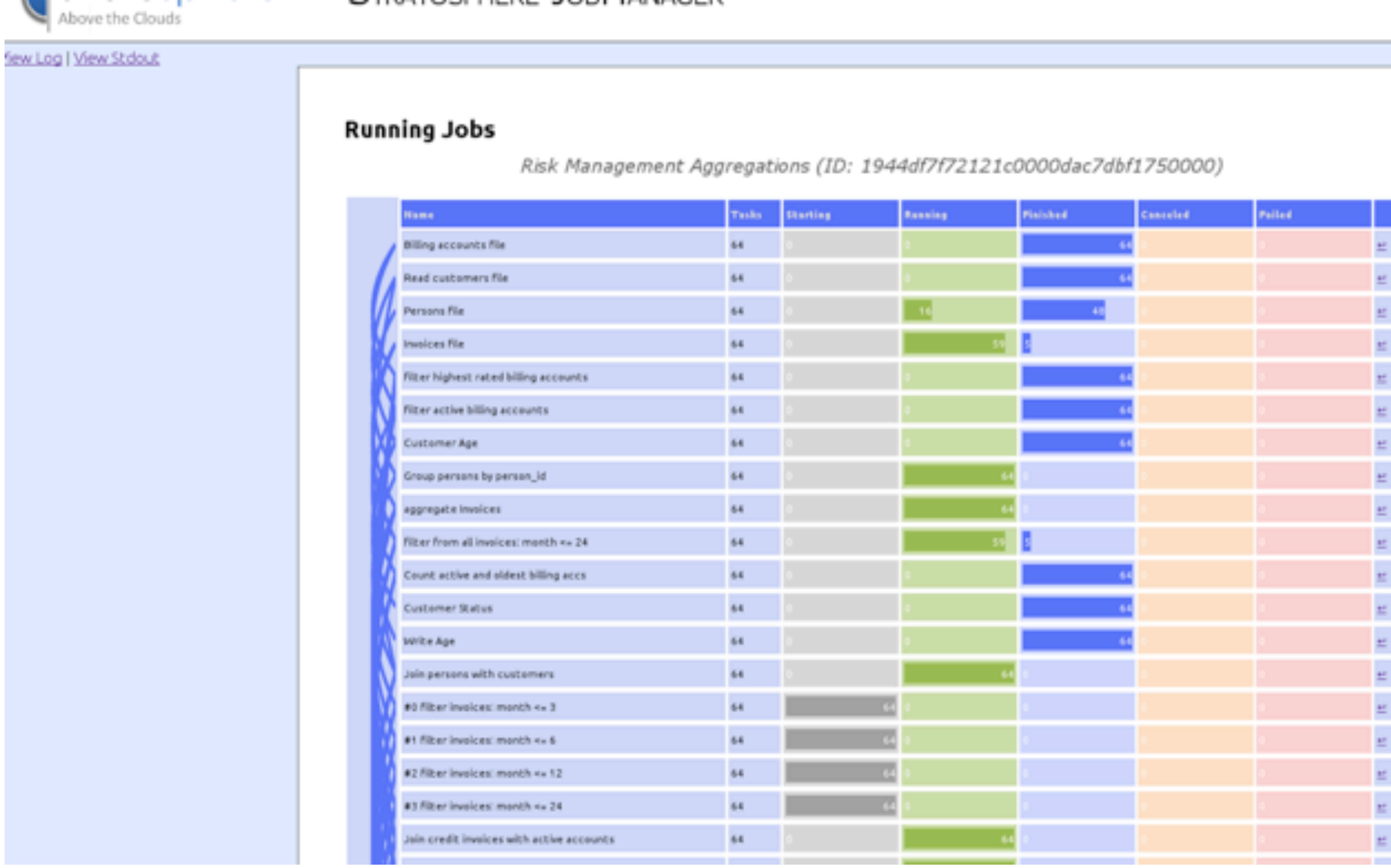
There are plenty of ways to explore Stratosphere. Install it one one or more machines, if you want to get to know the infrastructure. Application developers can also start immediately with their favorite programming language and run programs locally from within their favorite IDE.

Install Stratosphere on one or more computers.

Develop Stratosphere programs with Scala and experience Stratosphere's new concise and flexible programming abstraction. Run and debug your programs locally.

Write Stratosphere programs with the classic Java API. Run and debug your programs locally.

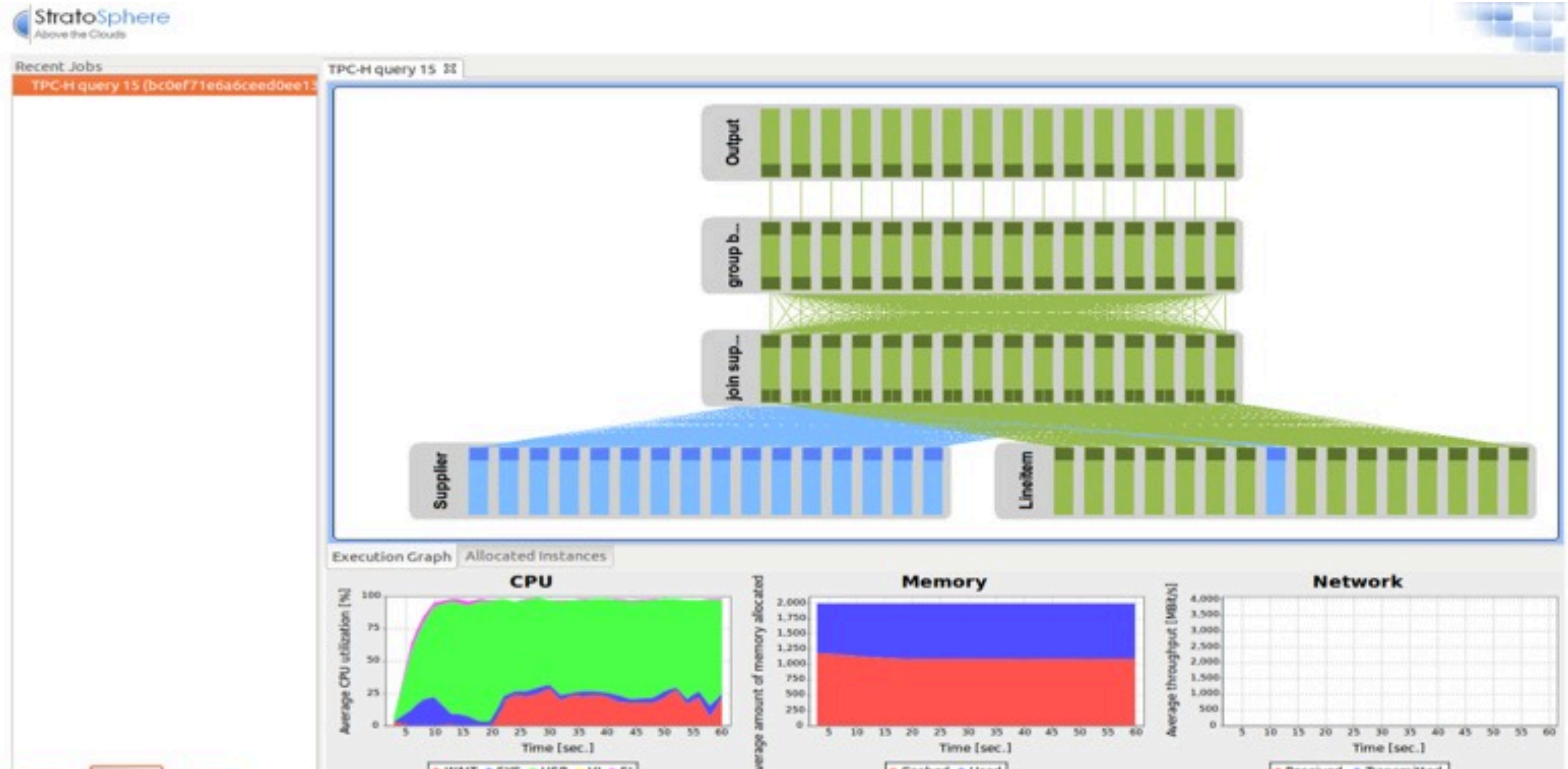
```
val input = TextFile(textInput)

val words = input
  .flatMap
       { line => line.split(“ “) }

val counts = words
  .groupBy
       { word => word }
  .count()

val output = counts
.write (wordsOutput,
        RecordDataSinkFormat() )

val plan = new ScalaPlan(Seq(output))
```



Running Jobs

Risk Management Aggregations (ID: 1944df7f72121c0000dac7dbf1750000)

Name	Tasks	Starting	Running	Finished	Canceled	Failed
Billing accounts file	64	0	0	64	0	0
Read customers file	64	0	0	64	0	0
Persons file	64	0	16	48	0	0
Invoices file	64	0	59	5	0	0
filter highest rated billing accounts	64	0	0	64	0	0
filter active billing accounts	64	0	0	64	0	0
Customer Age	64	0	0	64	0	0
Group persons by person_id	64	0	64	0	0	0
aggregate invoices	64	0	64	0	0	0
filter from all invoices: month <= 24	64	0	59	5	0	0
Count active and oldest billing accs	64	0	0	64	0	0
Customer Status	64	0	0	64	0	0
Write Age	64	0	0	64	0	0
Join persons with customers	64	0	64	0	0	0
#0 filter invoices: month <= 3	64	64	0	0	0	0
#1 filter invoices: month <= 6	64	64	0	0	0	0
#2 filter invoices: month <= 12	64	64	0	0	0	0
#3 filter invoices: month <= 24	64	64	0	0	0	0
Join credit invoices with active accounts	64	0	64	0	0	0



Help us shape the future of Big Data and the Stratosphere platform!

We are looking for contributions and pilot customers:

☛ github.com/stratosphere/stratosphere/wiki/Starter-Jobs

☛ Try out Stratosphere and give us feedback

☛ Work with us to implement your use case

Visit	**www.stratosphere.eu**
	www.github.com/stratosphere
Contact	**kostas.tzoumas@tu-berlin.de**
Tweet	**#StratoSummit**