Reverse Engineering Query Execution Engine Design Decisions

Helga Gudmundsdottir, Parmita Mehta,
Magdalena Balazinska, and Dan R. K. Ports

-- Work in Progress --


hadoop


Myria


MAPR

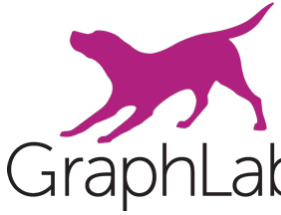

cassandra

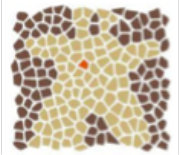

Spark


SHARK


SciDB
Developed by Paradigm4


Tokutek


GraphLab




Microsoft
SQL Server

PostgreSQL

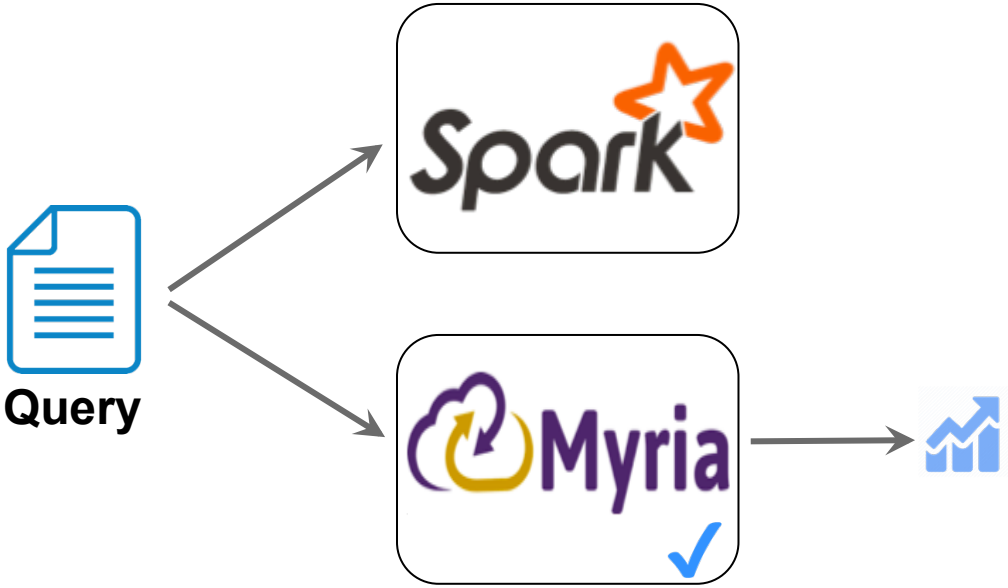

MySQL


APACHE
GIRAPH

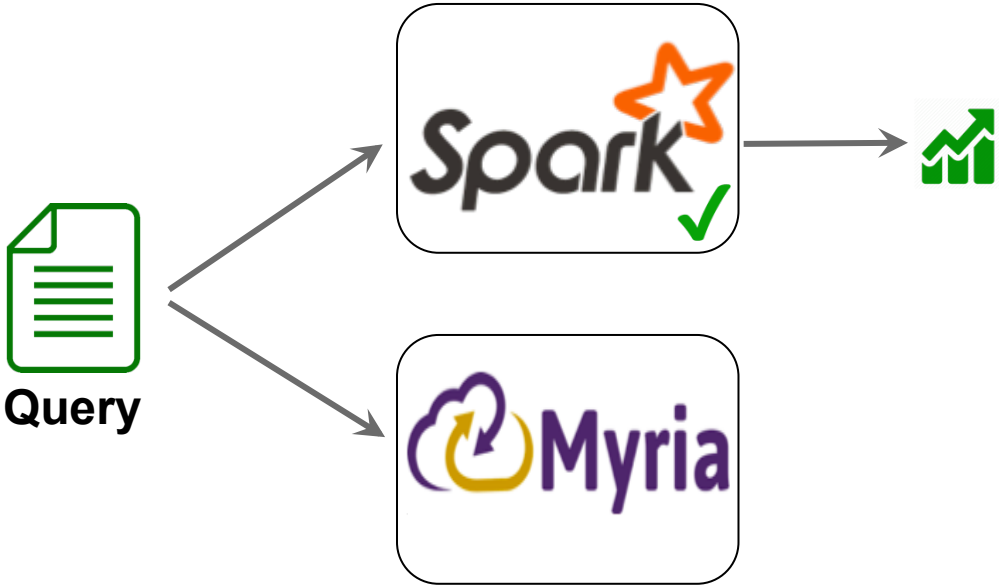

AKIBAN


The Datafloq Open Source Landscape 2.0
Data Analysis & Platforms
HPCC Systems
Storm
Hadoop
Dremel
hadoop map/reduce
Spark
HD
APACHE DRILL
SAMOA
IKANOW | BRILLIANT DECISIONS
Hortonworks
Databases / Data warehousing
bigdata
INFOBRIGHT
Cassandra
4store
H2
GlobalsDB
InfiniDB
riak
Infinispan
HYPERTABLE
MariaDB
Drizzle
SQLite
RethinkDB
terrastore
HIVE
Firebird
ORACLE
BERKELEY DB
HyperSQL
monetdb
In-Memory Computing
GridGain
IN-MEMORY COMPUTING
hazelcast
TERRACOTTA
GORA
NMemory
ERP BI Solutions
Business Intelligence
openi.org
Open Intelligence
talend
open data solutions
Jaspersoft
Palo
Open Source Business Intelligence
spagobi
pentaho
jedox.
BIRT
Data Mining
orange
rapidminer
KNIME
Data Analytics Made Easy
mahout
WEKA
KEEL
togaware
spmf
Big Data search
lucene
Apache
Solr
elasticsearch.
Multivalue database
Rocket
U2
REVELATION SOFTWARE
QM
Ladybridge Systems
northgate
jBASE INTERNATIONAL
ScarletDME
Programming
R
julia
Data aggregation
cascading
cloudera
Multidimensional
KeyValue
EROSPIKE
leveldb
redis
Chordless
Beta
Tokyo Cabinet
MEMCACHED
SCALIEN
Project Voldemort
A distributed database.
hamsterdb
RAPTORDB
FairCom
STSDB
DATABASE & VIRTUAL FILE SYSTEM
HyperDex
OpenLDAP
IQLECT
ioremap.net
STORAGE AND BEYOND
Scalaris
Document Store
Raven DB
mongoDB
CouchBase
CLUSTERPOINT
Tokutek
RaptorDB
EJDB
djondb
JasDB
SchemafreeDB
sisodb
Apache CouchDB
relax
Graph databases
Gephi
Gremlin
GraphBuilder
FRANZ INC.
*Sparksee
INFO GRID
InfiniteGraph
HYPERGRAPHDB
Neo4j
FlockDB
GraphBase
BrightstarDB
Operational
VOLTDB
Socia
Apach
ThinkU
db4objects
Object databases
ZOPE
mobject
Magma
Picolisp
NEOPPOD
siaqodb
FRAMER D
PERSEVERE
EyeDB
STARCOUNTER
Sterling
NDatabase
C# Lightweight Object Database
Multimodel
ArangoDB
alchemydatabase
XML Database
existdb
BASE
sedna
LIQU
3

Same analytics query on two different systems



Different analytics query on same two systems



What are these systems really doing?

- Writing intermediate data to disk? Vs in-memory?
- CPU-intensive operations?
- Shuffling data?
- Inefficient memory management?
- Any synchronization barriers? Data skew?
- Able to use entire cluster?
- Were there failures?
- ...

Want to understand the underlying design decisions of query execution engines

Requires:

- Deep expertise in each DBMS
- Detailed experimentation
- Knowledge of code base
- Instrumentation & profiling


hadoop


Myria


MAPR


cassandra

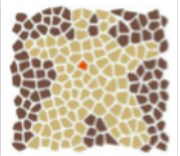

Spark


SHARK


SciDB
Developed by Paradigm4


Tokutek


GraphLab




Microsoft
SQL Server

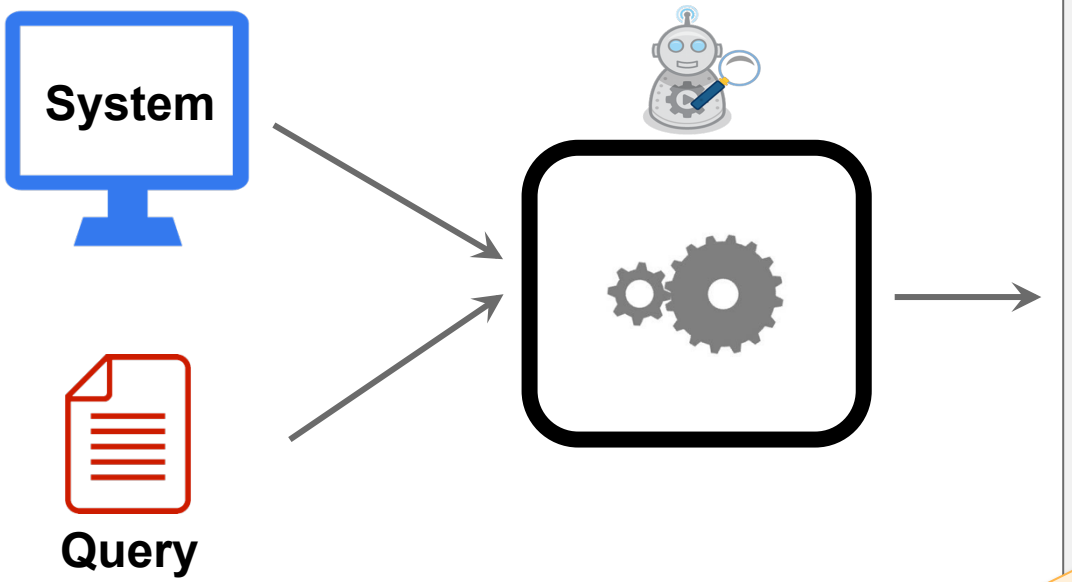

PostgreSQL


MySQL


APACHE
GIRAPH


AKIBAN

Is there a way to do this automatically?



- ➔ 80% workers utilized
- ➔ Three data shuffling ops
- ➔ No synch. barriers
- ➔ Periodic data materialization
- ➔ Column-store storage

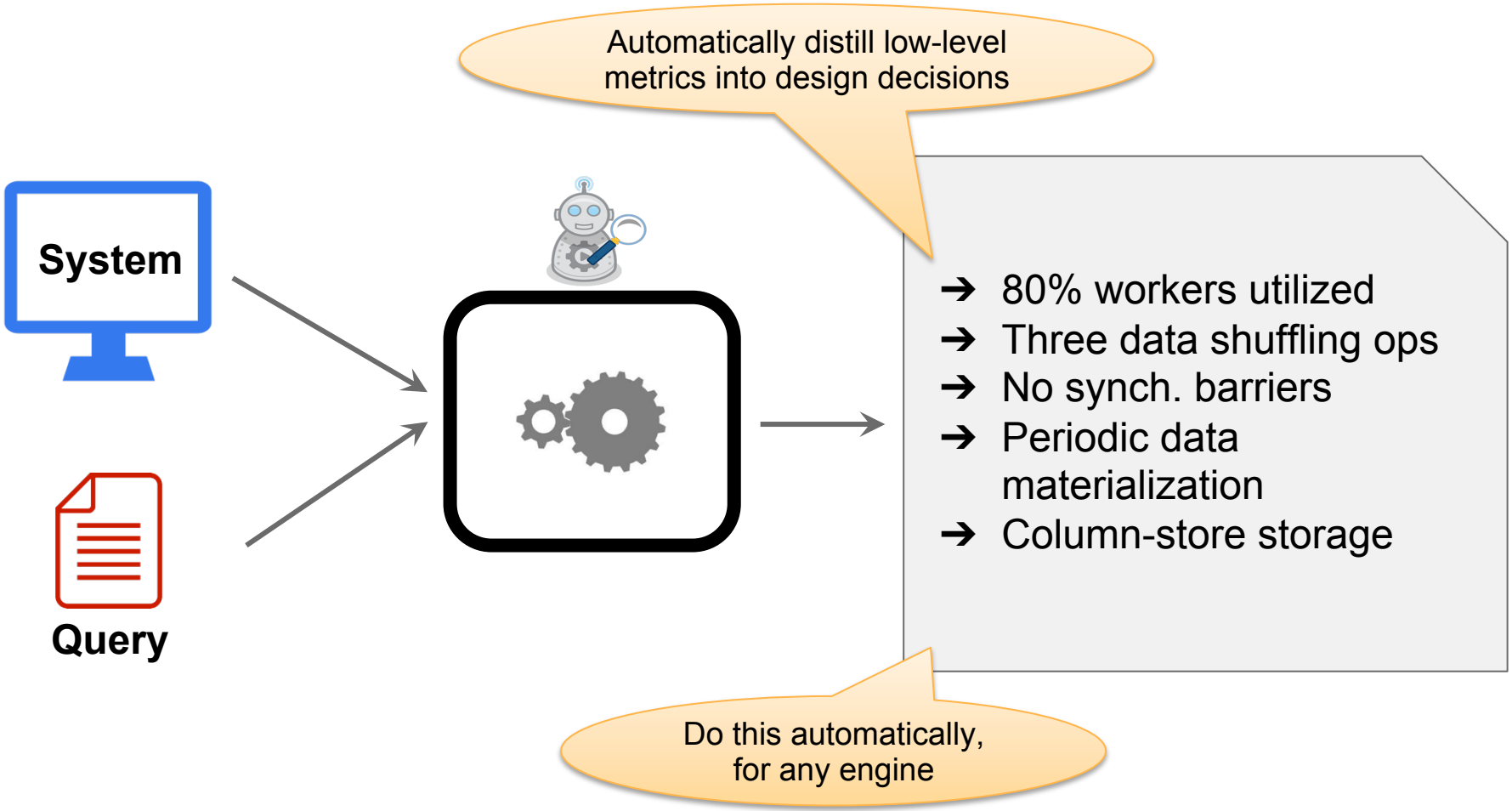
Do this automatically, for any engine

How do design decisions manifest in low-level system metrics?

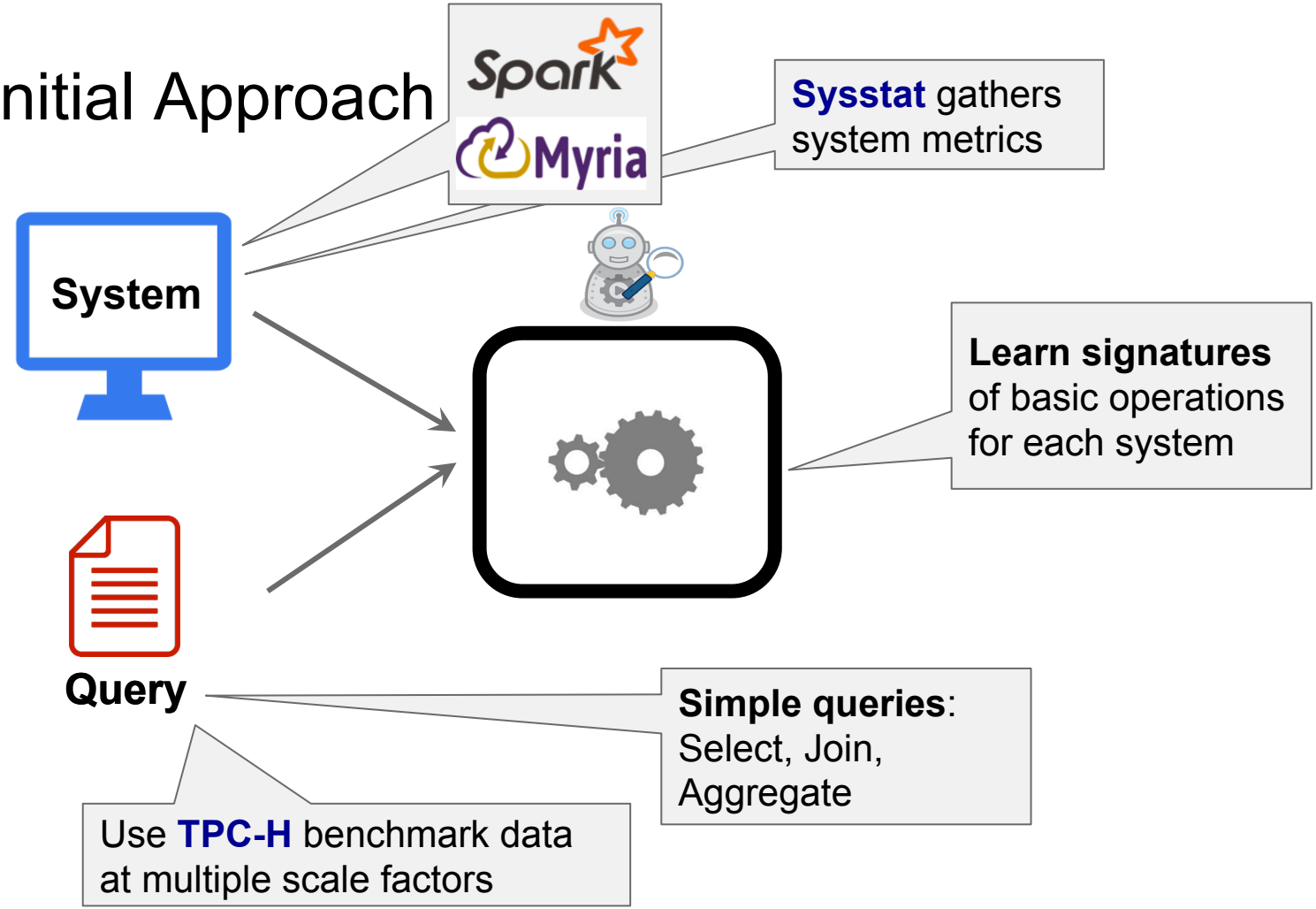
- Can we learn their signatures?

Can system metrics provide insights into how design decisions manifest during query execution?

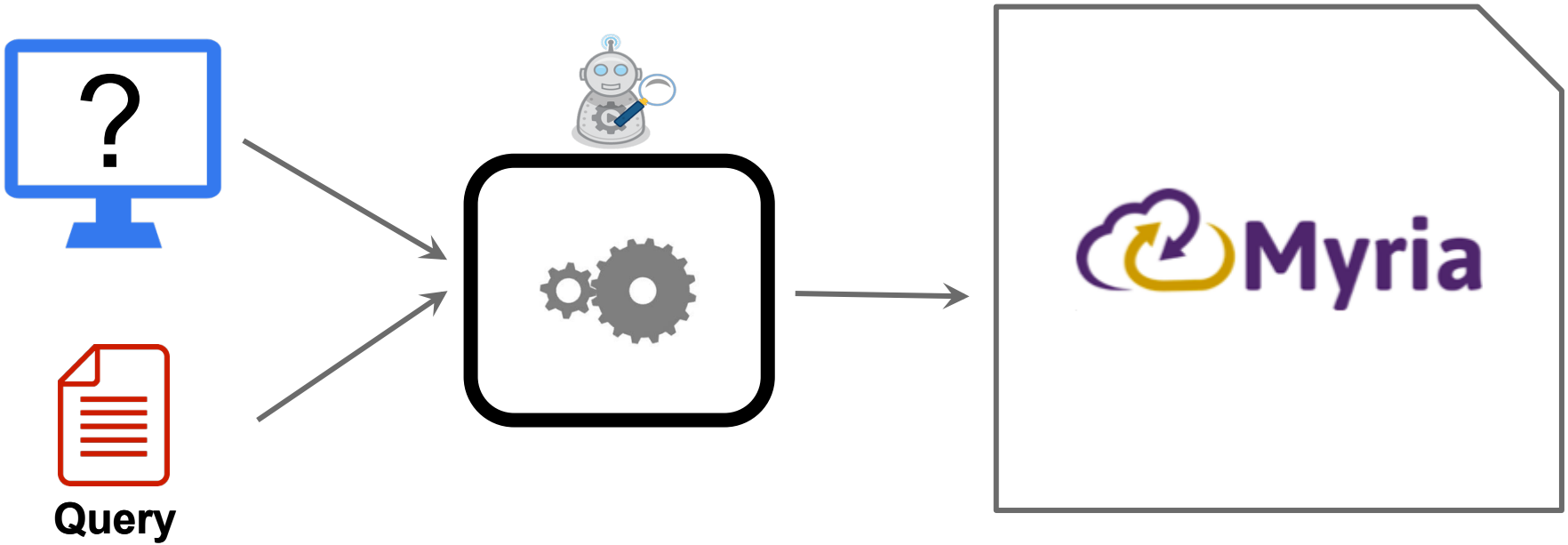
- Provide rich information about query execution and system behavior


Automatically distill low-level metrics into design decisions
System
Query
➔ 80% workers utilized
➔ Three data shuffling ops
➔ No synch. barriers
➔ Periodic data materialization
➔ Column-store storage
Do this automatically, for any engine

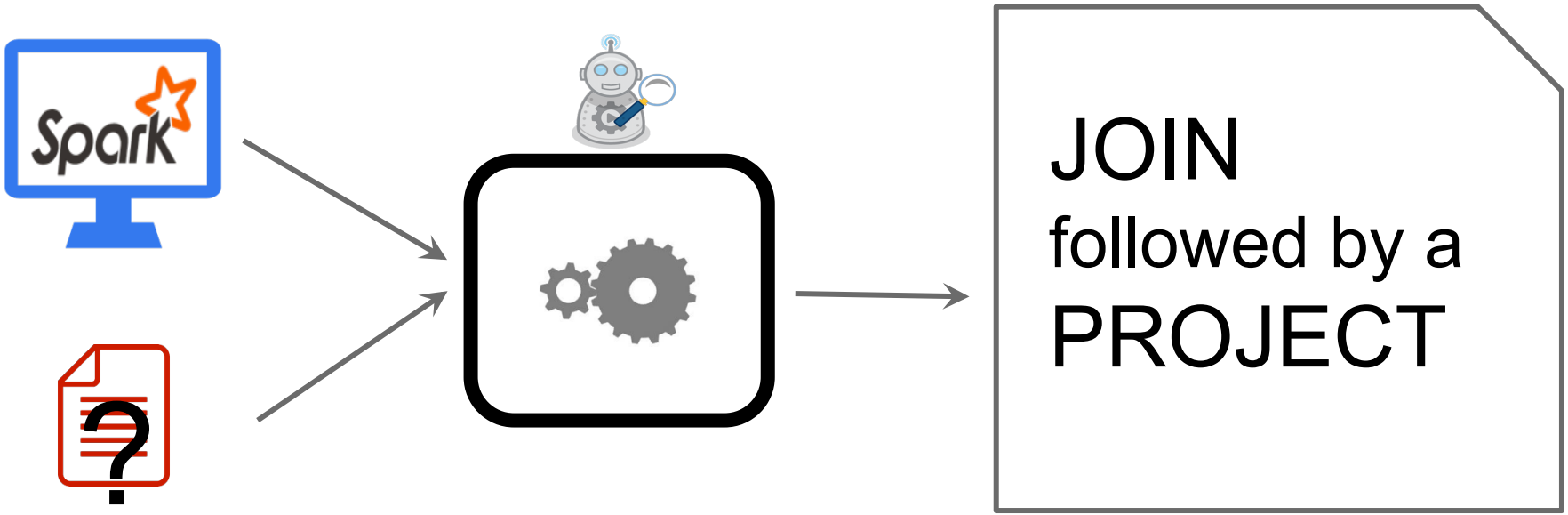
Initial Approach



Can we distinguish between systems?



Can we distinguish between basic operators?



Questions to the audience

- Would you use such a system?
- What design decisions are worth inferring?
- Other applications of this approach?

SLA for Multi-tenant Analytics as a Service

Analytics as a Service

Dedicated resources can provide performance guarantees

Over-provisioning can provide predictable response times

But, what if we want analytics cheaper?

SLA for Multi-tenant Analytics Service

Can the information about system load be captured and displayed to customers utilizing the system?

How to expose contention that is interpretable by the user?

- Slow down factor for the service?
- How long before query will complete?

SLA for Multi-tenant Analytics Service

What does the user want?

- How should we think about SLAs in Analytics as a service with contention?
- How should we expose this information to end users?
- What scheduling algorithms make sense for Analytics as a service?