BlockchainOnAkka **finds** IDRs
(without Akka)

Step 1: Read the paper.
Step 2: Learning about Spark's quirks by doing.
Step 3: ???
Step 4: Profit.

Implementation details:

- Encode all data as Strings.
- Encode column names with Integers (saved ~10% of the time).

Scaling Out the Discovery of Inclusion Dependencies

Sebastian Kruse, Thorsten Papenbrock, Felix Naumann

Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam
firstname.lastname@hpi.de

Abstract: Inclusion dependencies are among the most important database dependencies. In addition to their most prominent application – foreign key discovery – inclusion dependencies are an important input to data integration, query optimization, and schema redesign. With their discovery being a recurring data profiling task, previous research has proposed different algorithms to discover all inclusion dependencies within a given dataset. However, none of the proposed algorithms is designed to scale out, i.e., none can be distributed across multiple nodes in a computer cluster to increase the performance. So on large datasets with many inclusion dependencies, these algorithms can take days to complete, even on high-performance computers.

We introduce SINDY, an algorithm that efficiently discovers all unary inclusion dependencies of a given relational dataset in a distributed fashion and that is not tied to main memory requirements. We give a practical implementation of SINDY that builds upon the map-reduce-style framework Stratosphere and conduct several experiments showing that SINDY can process huge datasets by several factors faster than its competitors while scaling with the number of cluster nodes.