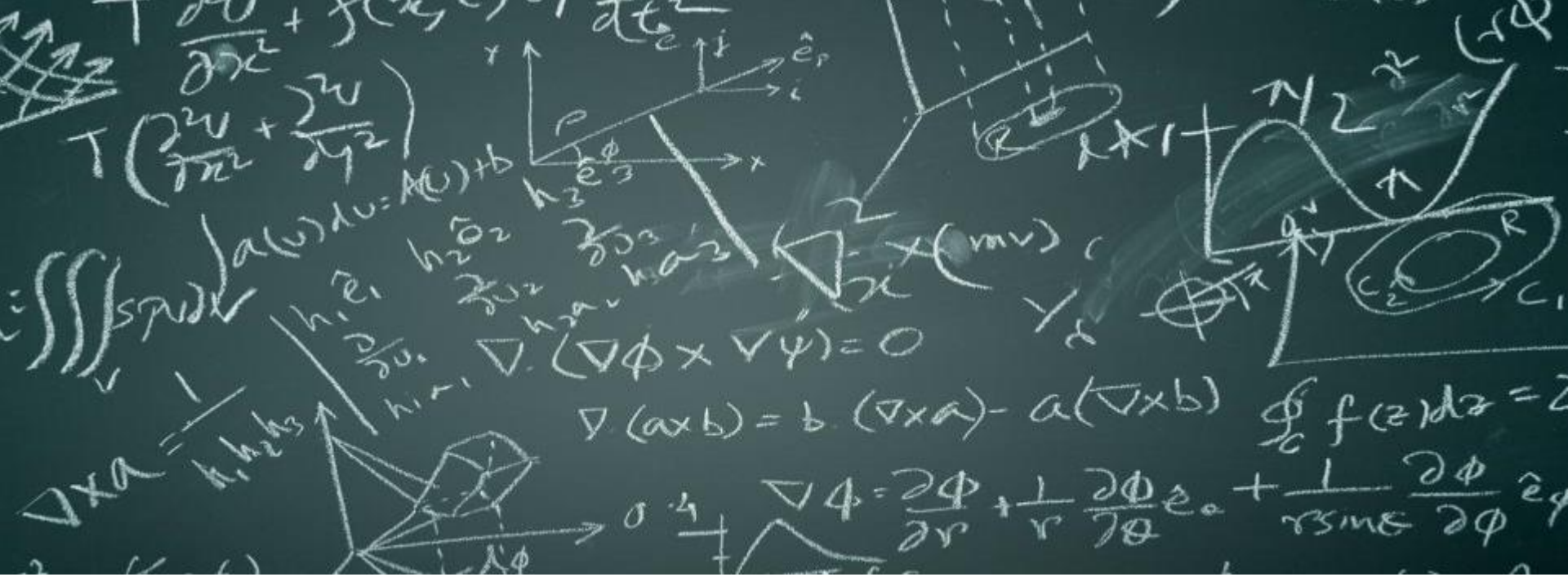


Big Data Analytics with the HPCC Systems Platform

Data and Visual Analytics
Prof. Duen Horng (Polo) Chau

Georgia Institute of Technology
Presenter: Dr. Flavio Villanustre
Lead of the HPCC Systems Initiative

LexisNexis

hpccsystems.com

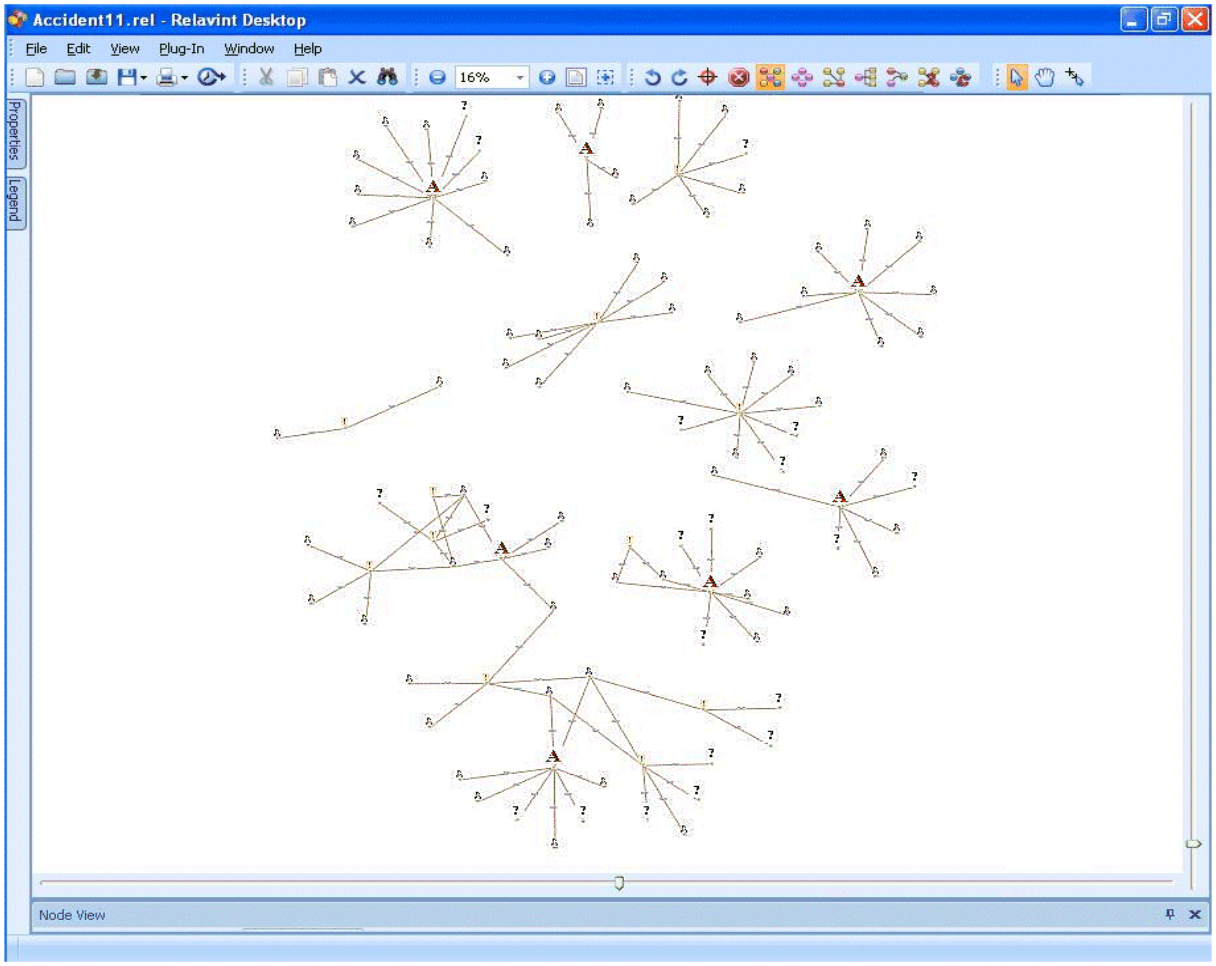
Real Life Graph Analytics

HPCC Systems
http://hpccsystems.com

Scenario

This view of carrier data shows seven known fraud claims and an additional linked claim.

The Insurance company data **only finds a connection between two of the seven claims**, and only identified one other claim as being weakly connected.



Real Life Graph Analytics

HPCC Systems
http://hpccsystems.com

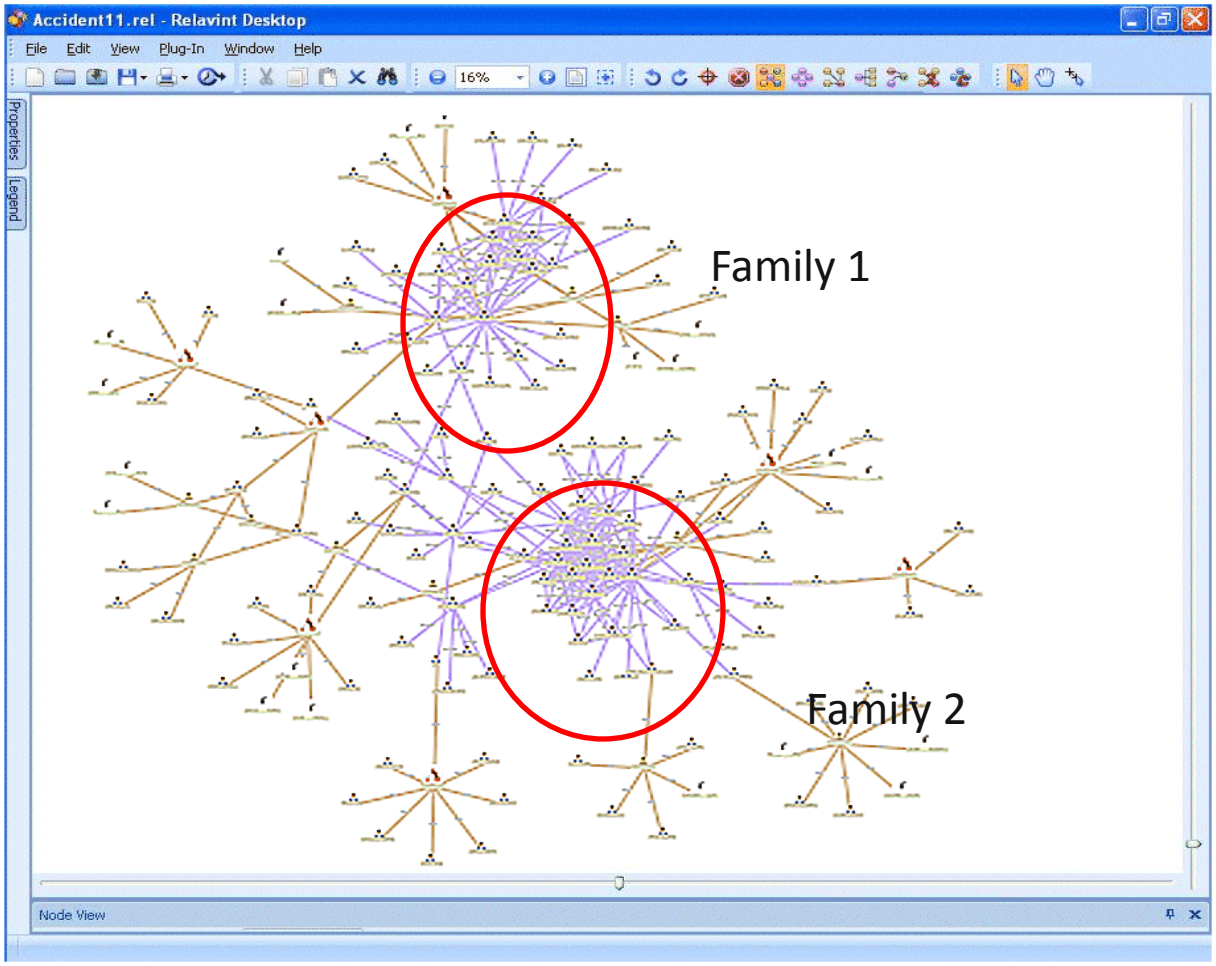
Task

After adding the LexID to the carrier Data, LexisNexis HPCC technology then explored 2 additional degrees of relative separation

Result

The results showed **two family groups interconnected on all of these seven claims**.

The links were much stronger than the carrier data previously supported.

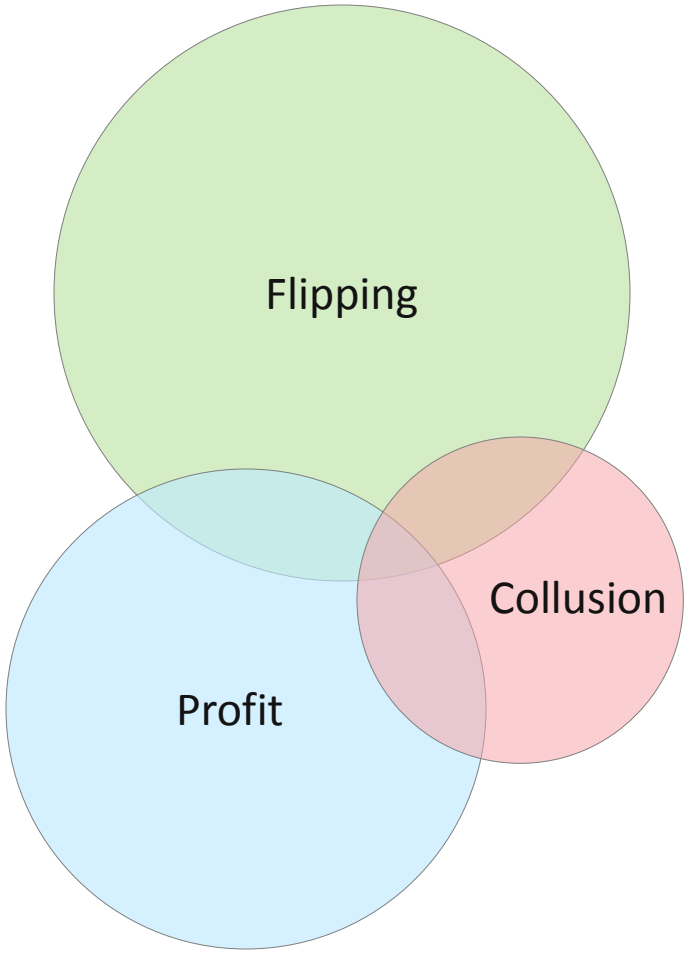


Property Transaction Risk

HPCC Systems
http://hpccsystems.com

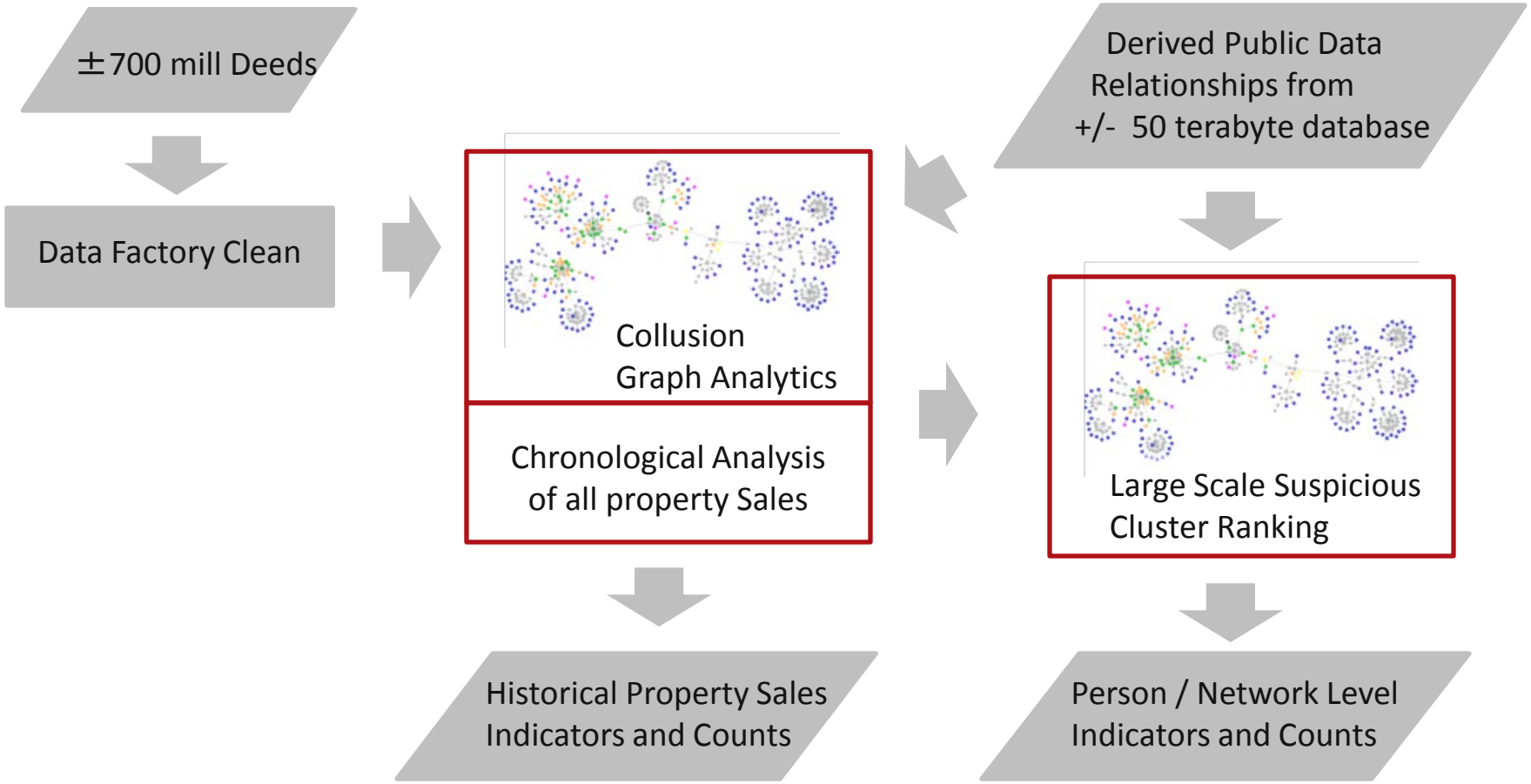
Three core transaction variables measured

- Velocity
- Profit (or not)
- Buyer to Seller Relationship Distance (Potential of Collusion)



Property Transaction Risk

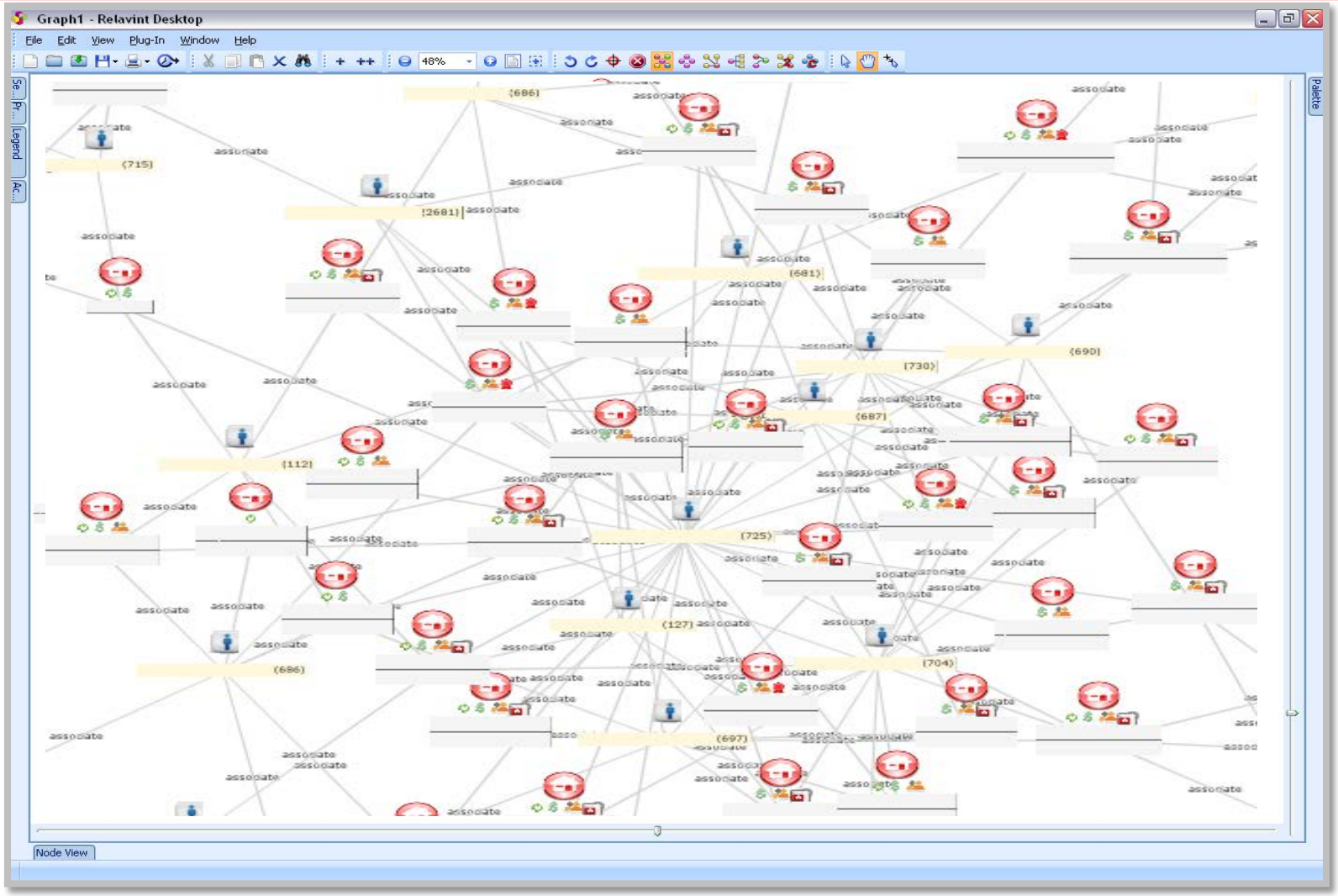
HPCC Systems
http://hpccsystems.com



HPCC Systems

Suspicious Equity Stripping Cluster

HPCC Systems
http://hpccsystems.com



HPCC Systems

Results

HPCC Systems
http://hpccsystems.com

Large scale measurement of influencers strategically placed to potentially direct suspicious transactions.

- All BIG DATA on one supercomputer measuring over a decade of property transfers nationwide.
- BIG DATA Products to turn other BIG DATA into compelling intelligence.
- Large Scale Graph Analytics allow for identifying known unknowns.
- Florida Proof of Concept
  - Highest ranked influencers
    - Identified known ringleaders in flipping and equity stripping schemes.
    - Typically not connected directly to suspicious transactions.
  - Known ringleaders not the Highest Ranking.
- Clusters with high levels of potential collusion.
- Clusters offloading property, generating defaults.
- Agile Framework able to keep step with emerging schemes in real estate.

Social Graph and Prescriptions

HPCC Systems
http://hpccsystems.com

Scenario

Healthcare insurers need better analytics to identify drug seeking behavior and schemes that recruit members to use their membership fraudulently.

Groups of people collude to source schedule drugs through multiple members to avoid being detected by rules based systems.

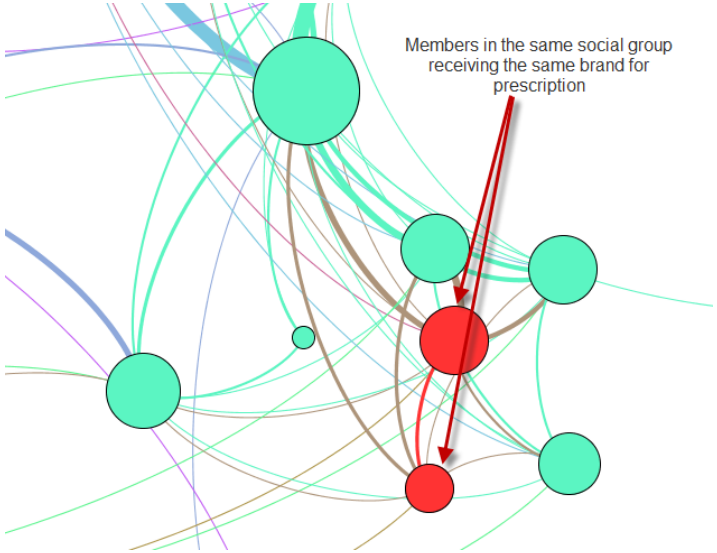
Providers recruit members to provide and escalate services that are not rendered.

Task

Given a large set of prescriptions. Calculate normal social distributions of each brand and detect where there is an unusual socialization of prescriptions and services.

Result

The analysis detected social groups that are sourcing Vicodin and other schedule drugs. Identifies prescribers and pharmacies involved to help the insurer focus investigations and intervene strategically to mitigate risk.



Network Traffic Analysis in Seconds

HPCC Systems
http://hpccsystems.com

Scenario

Conventional network sensor and monitoring solutions are constrained by inability to quickly ingest massive data volumes for analysis

- 15 minutes of network traffic can generate 4 Terabytes of data, which can take 6 hours to process
- 90 days of network traffic can add up to 300+ Terabytes

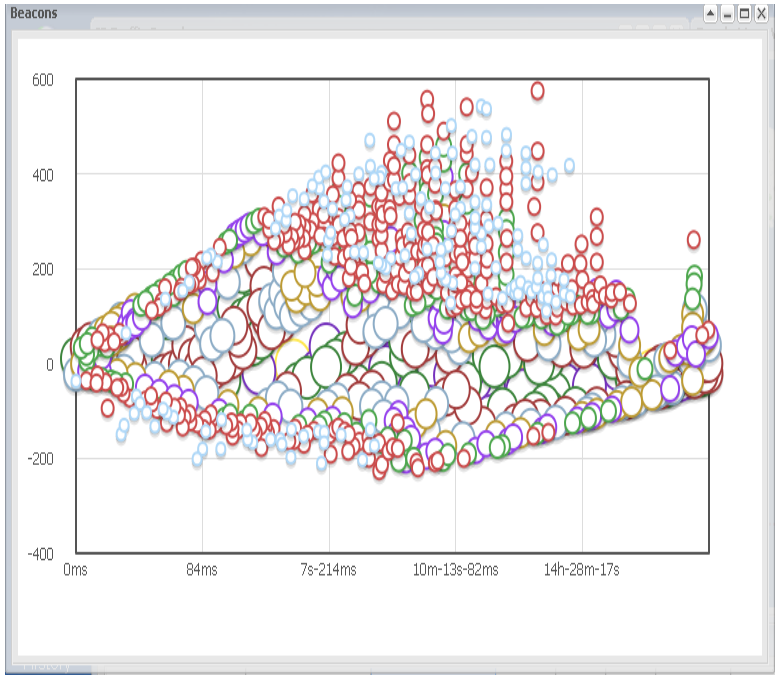
Task

Drill into all the data to see if any US government systems have communicated with any suspect systems of foreign organizations in the last 6 months

- In this scenario, we look specifically for traffic occurring at unusual hours of the day

Result

In seconds, the HPCC sorted through months of network traffic to identify patterns and suspicious behavior



Horizontal axis:	time on a logarithmic scale
Vertical axis:	standard deviation (in hundredths)
Bubble size:	number of observed transmissions

Wikipedia Graph Demo

HPCC Systems
http://hpccsystems.com

Scenario

Calculate Google Page Rank to be used to rank search results and drive visualizations.

Task

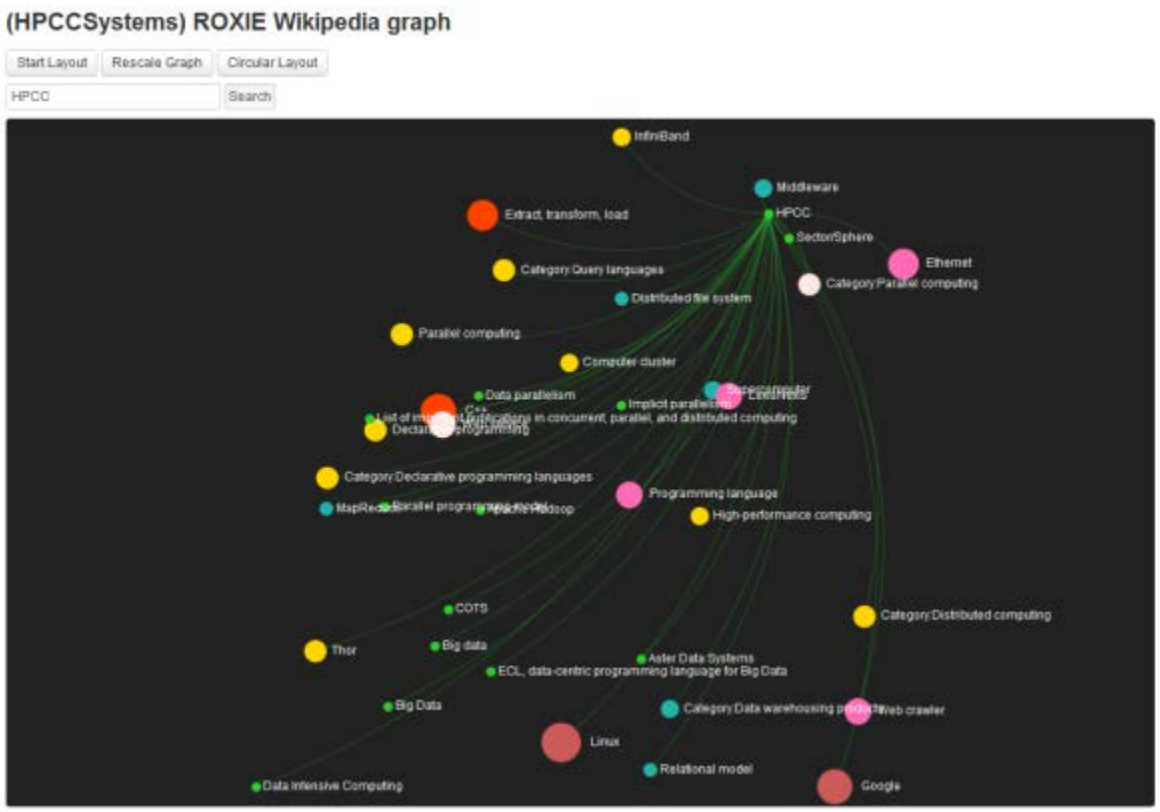
Load the 75GIG English Wikipedia XML snapshot. Strip page links from all pages and run 20 iterations of Google Page Rank Algorithm. Generate indexes and Roxie query to support visualization.

Result

Produces +- 300 million links between 15 million pages. Page Rank allows for ranking results in searching and driving more intuitive visualizations.

Advanced

Lays a foundation for advanced graph algorithms that combine ranking, Natural Language Processing and Machine Learning in scale.



Interactive Visualization uses calculated page rank to define size and color of the nodes.

Wikipedia Pageview Demo

HPCC Systems
http://hpccsystems.com

Scenario

21 Billion Rows of Wikipedia Hourly Page view Statistics for a year.

Task

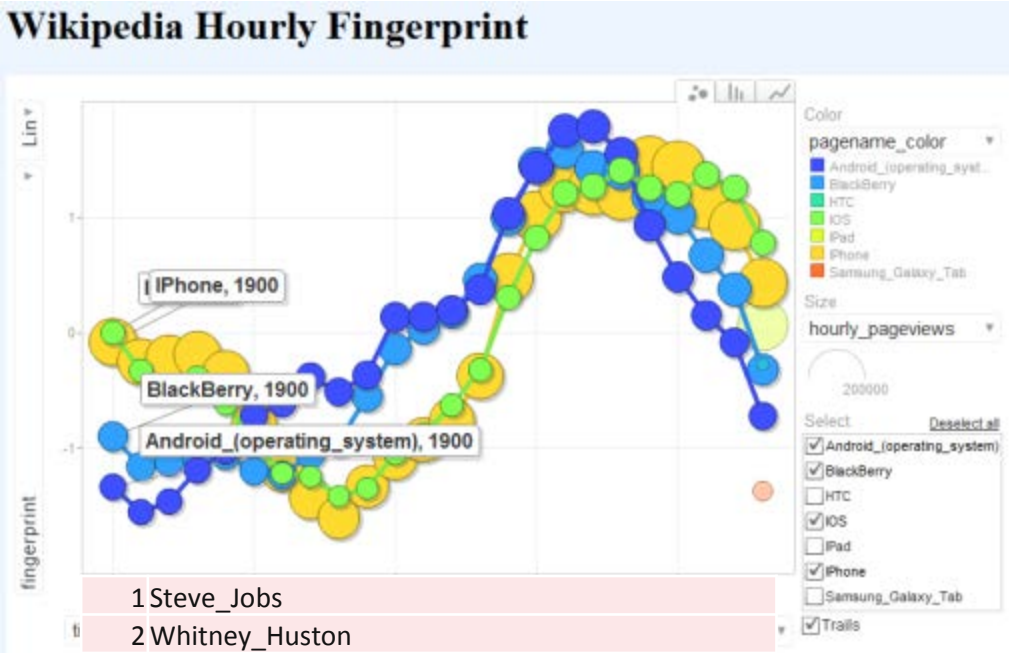
Generate meaningful statistics to understand aggregated global interest in all Wikipedia pages across the 24 hours of the day built off all English Wikipedia page view logs for 12 months.

Result

Produces page statistics that can be queried in seconds to visualize which key times of day each Wikipedia page is more actively viewed.

Helps gain insight into both regional and time of day key interest periods in certain topics.

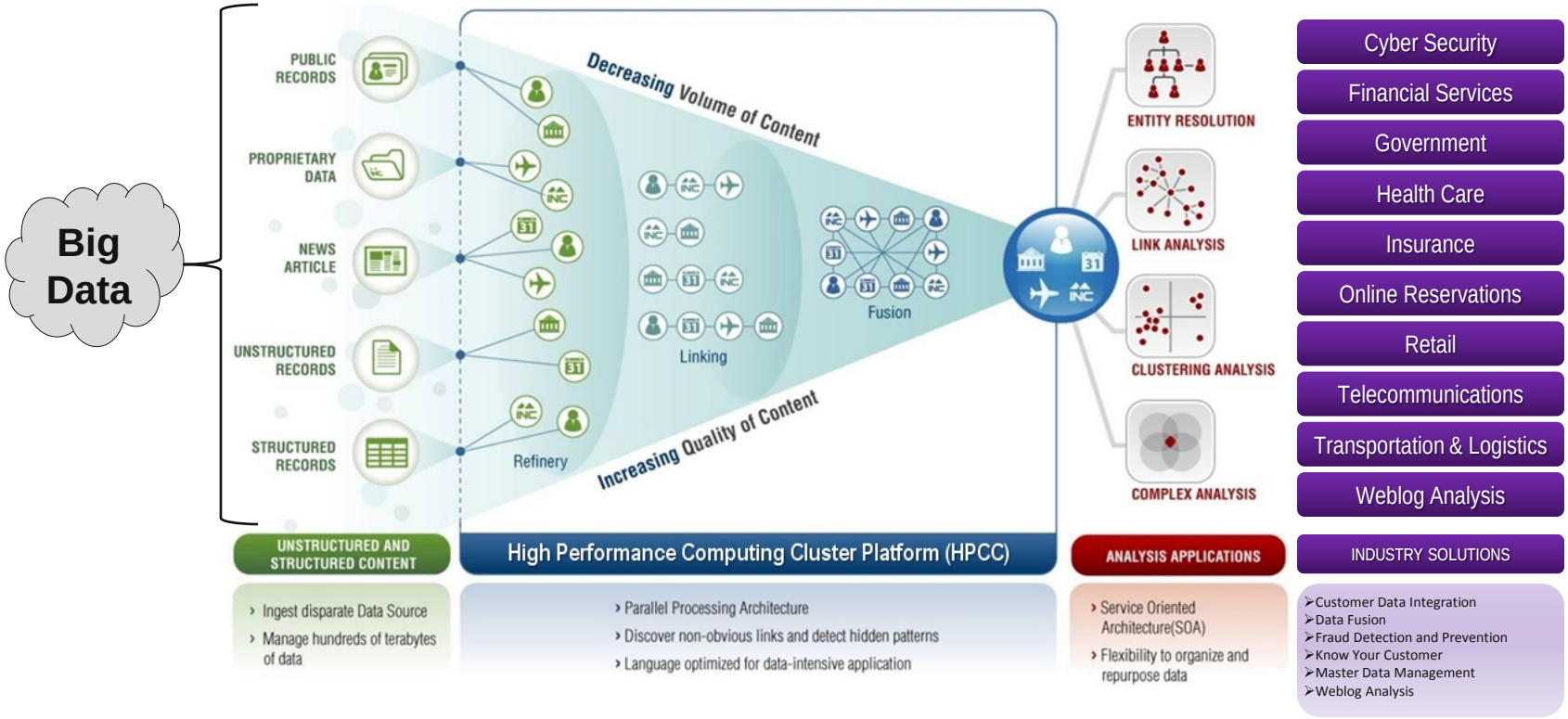
This result can be leveraged with Machine Learning to cluster pages with similar 24hr Fingerprints.



1	Steve_Jobs
2	Whitney_Huston
3	Wikipedia:SOPA_initiative/Learn_more
4	List_of_HTTP_status_codes%231xx_Informational
5	Adele_(singer)
6	Bruno_Mars
7	Kim_Jong-il
8	Jeremy_Lin
9	Heavy_D
10	Christopher_Columbus
11	Murder_of_Meredith_Kercher
12	Eli_Manning
13	Jorge_Luis_Borges

The Data/Information flow

HPCC Systems
http://hpccsystems.com



- **High Performance Computing Cluster Platform (HPCC)** enables data integration on a scale not previously available and real-time answers to millions of users. Built for big data and proven for 10 years with enterprise customers.
- **Offers a single architecture**, two data platforms (query and refinery) and a consistent data-intensive programming language (ECL)
- **ECL Parallel Programming Language** optimized for business differentiating data intensive applications

The Open Source HPCC Systems platform

HPCC Systems
http://hpccsystems.com

- Open Source distributed data-intensive computing platform
- Shared-nothing architecture
- Runs on commodity computing/storage nodes
- Binary packages available for the most common Linux distributions
- Provides end-to-end Big Data workflow management , scheduler, integration tools, etc.
- Originally designed circa 1999 (predates the original paper on MapReduce from Dec. ‘04)
- Improved over a decade of real-world Big Data analytics
- In use across critical production environments throughout LexisNexis for more than 10 years

Components

- The HPCC Systems platform includes:
  - Thor: batch oriented data manipulation, linking and analytics engine
  - Roxie: real-time data delivery and analytics engine
- A high level declarative data oriented language: ECL
  - Implicitly parallel
  - No side effects
  - Code/data encapsulation
  - Extensible
  - Highly optimized
  - Builds graphical execution plans
  - Compiles into C++ and native machine code
  - Common to Thor and Roxie
- An extensive library of ECL modules, including data profiling, linking and Machine Learning

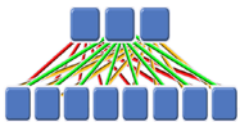
The Three HPCC components

HPCC Systems
http://hpccsystems.com

1 **HPCC Data Refinery (Thor)**

- Massively Parallel Extract Transform and Load (ETL) engine
- Enables data integration on a scale not previously available
- Programmable using ECL

2 **HPCC Data Delivery Engine (Roxie)**



- A massively parallel, high throughput, query engine
- Low latency, highly concurrent and highly available
- Allows multipart (compound) indices for efficient retrieval
- Programmable using ECL

3 **Enterprise Control Language (ECL)**

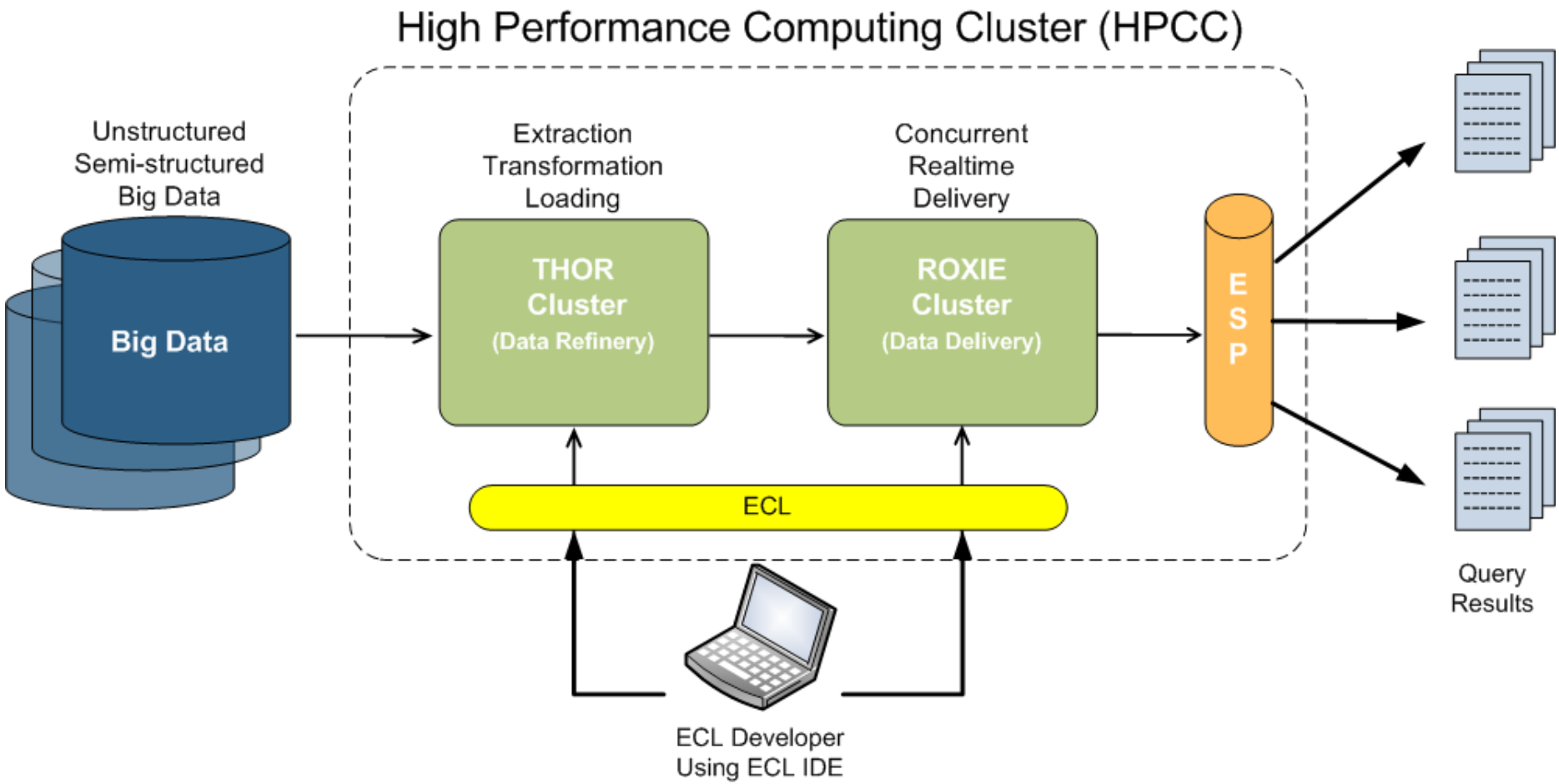
- An easy to use, declarative data-centric programming language optimized for large-scale data management and query processing
- Highly efficient; automatically distributes workload across all nodes.
- Automatic parallelization and synchronization

Conclusion: End to End platform

- No need for any third party tools

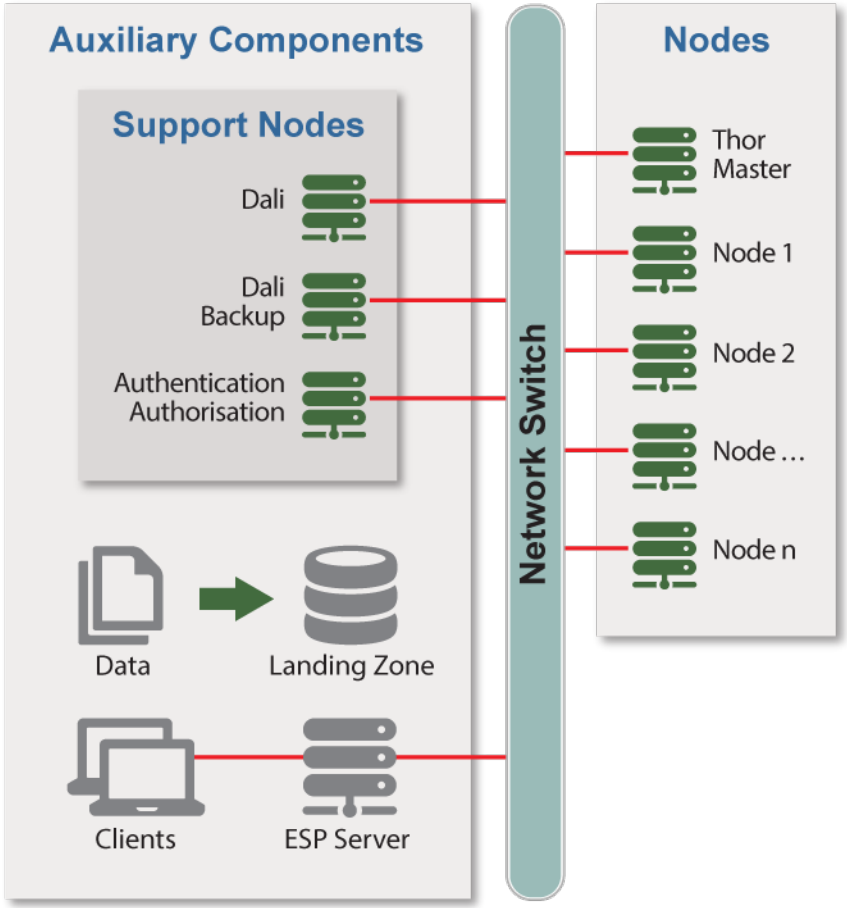
The HPCC Systems platform

HPCC Systems
http://hpccsystems.com

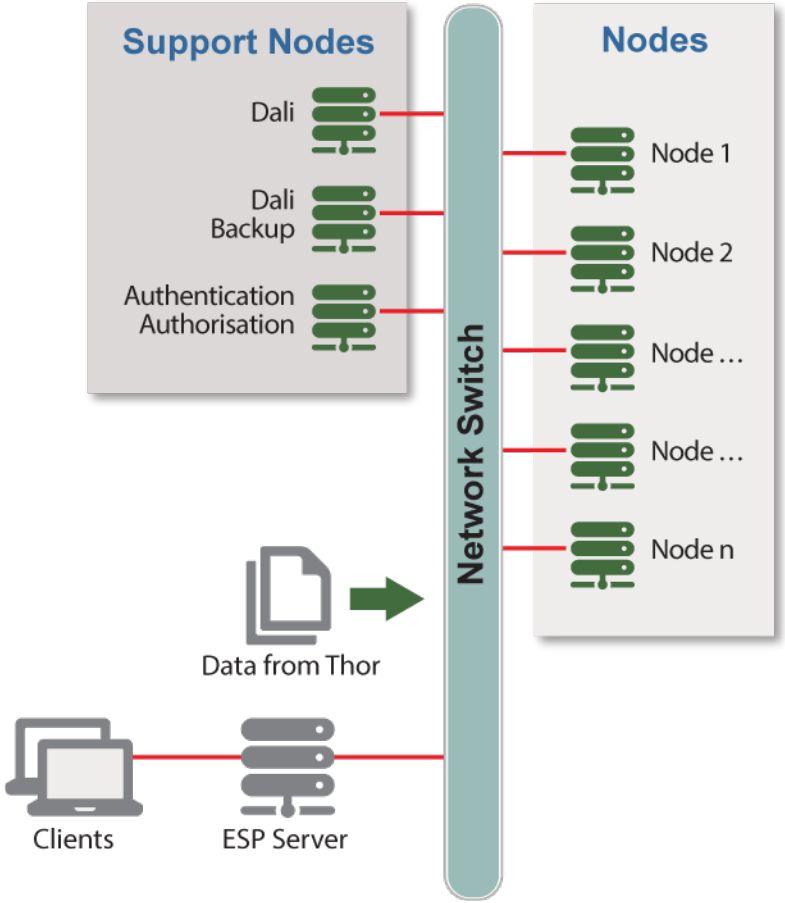


Detailed HPCC Architecture

HPCC Systems
http://hpccsystems.com



Thor
(Batch Job Execution Engine + DFS)
Physical Layout Schematic Diagram



Roxie
(Rapid Data Delivery Engine)
Physical Layout Schematic Diagram

Enterprise Control Language (ECL)

HPCC Systems
http://hpccsystems.com

Declarative programming language: Describe **what** needs to be done and **not how** to do it

Powerful: High level data primitives as JOIN, TRANSFORM, PROJECT, SORT, DISTRIBUTE, MAP, etc. are available.

Extensible: As new attributes are defined, they become primitives that other programmers can use

Implicitly parallel: Parallelism is built into the underlying platform. The programmer needs not be concerned with it

Maintainable: A high level programming language, without side effects and with efficient encapsulation, programs are more succinct, reliable and easier to troubleshoot

Complete: ECL provides for a complete data programming paradigm

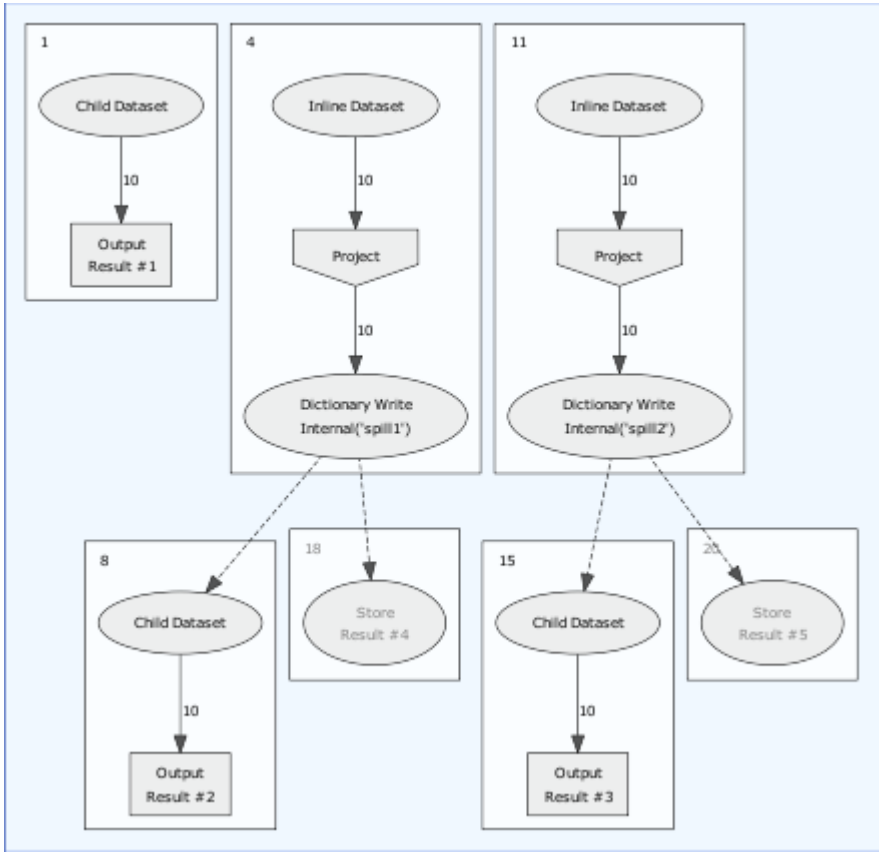
Homogeneous: One language to express data algorithms across the entire HPCC platform, data integration, analytics and high speed delivery

```
// Initialize output log
log_out_init := project(log_init,
                        transform(layout_logo
                        self := left,
                  self := []));

// Create error log
outerrorfile := join(log_seq,
                     log_out_init,
           left.linenum = right.linenum,
           transform(recordof(log_seq),
                     self := left),
           left only,
           hash);

// Denormalize key value pairs
outlogfile := sort(denormalize(distribute(log_
                           sort(distribute(key
                left.linenum = right.linenum,
                transform(layout_logout,
                          self.keyvals := left
                                          row({ric
                     self := left),
```

Enterprise Control Language (ECL)

HPCC Systems
http://hpccsystems.com

- The ECL optimizer generates an optimal execution plan
- Subgraphs run activities in parallel, automatically persist the results and spill to disk as needed
- Lazy evaluation avoids re-computing results when data and code haven't changed
- Graphs are dynamic and display the number of records traversing the edges and data skew percentages



Machine Learning on HPCC

HPCC Systems
http://hpccsystems.com

- Extensible Machine Learning Library developed in ECL
- Fully distributed across the cluster
- Supports PB-BLAS (Parallel Block BLAS)
- General statistical functions
- Supports supervised, semi-supervised and unsupervised learning methods
- Document manipulation, tokenization and statistical Natural Language Processing
- A consistent and standard interface to classification ("pluggable classifiers")
- Efficient handling of iterative algorithms (for example, for numeric optimization and clustering)
- Open Source and available at: http://hpccsystems.com/ml

ECL-ML: extensible ML on HPCC

HPCC Systems
http://hpccsystems.com

General aspects

- Based on a distributed ECL linear algebra framework
- New algorithms can be quickly developed and implemented
- Common interface to classification (pluggable classifiers)

ML algorithms

- Linear regression
- Several Classifiers
- Multiple clustering methods
- Association analysis

Document manipulation and statistical grammar-free NLP

- Tokenization
- CoLocation

Statistics

- General statistical methods
- Correlation
- Cardinality
- Ranking

ECL-ML: extensible ML on HPCC (ii)

Linear Algebra library

- Support for sparse matrices
- Standard underlying matrix/vector data structures
- Basic operations (addition, products, transpositions)
- Determinant/inversions
- Factorization/SVD/Cholesky/Rank/UL
- PCA
- Eigenvectors/Eigenvalues
- Interpolation (Lanczos)
- Identity
- Covariance
- KD Trees

Machine Learning on HPCC

HPCC Systems
http://hpccsystems.com

- ML on a general-purpose Big Data platform means effective analytics in-situ
- The combination of Thor and Roxie is ideal when, for example, training a model on massive amounts of labeled historical records (Thor), and providing real-time classification for new unlabeled data (Roxie)

REMEMBER!

When applying Machine Learning methods to Big Data: data profiling, parsing, cleansing, normalization, standardization and feature extraction represent 85% of the problem!

The Future: Knowledge Engineering

HPCC Systems
http://hpccsystems.com

At the end of the day

- Do I really care about the format of the data?
- Do I even care about the placement of the data?
- I do care (a lot!) about what can be inferred from the data
- The context is important as long as it affects my inference process
- I want to leverage existing algorithms

ECL	KEL
Generates C++ (1->100)	Generates ECL (1->12)
Files and Records	Entities and associations
Detailed control of data format	Loose control of input format; none of processing
Can write graph and statistical algorithms	Major algorithms built in
Thor/Roxie split by human design	Thor/Roxie split by system design
Solid, reliable and mature	R&D

KEL by example (WIP!)

HPCC Systems
http://hpccsystems.com

- Actor := ENTITY(FLAT(UID(ActorName),Actor=ActorName))
- Movie := ENTITY(FLAT(UID(MovieName),Title=MovieName))
- Appearance := ASSOCIATION(FLAT(Actor Who,Movie What))

- USE IMDB.File_Actors(FLAT,Actor,Movie,Appearance)

- CoStar := ASSOCIATION(FLAT(Actor Who,Actor WhoElse))

- GLOBAL: Appearance(#1,#2) Appearance(#3,#2) => CoStar(#1,#3)

- QUERY:FindActors(_Actor) <= Actor(_Actor)
- QUERY:FindMovies(_Actor) <= Movie(UID IN Appearance(Who IN Actor(_Actor){UID}){What})
- QUERY:FindCostars(_Actor) <= Actor(UID IN CoStar(Who IN Actor(_Actor){UID}){WhoElse})
- QUERY:FindAll(_Actor) <= Actor(_Actor),Movie(UID IN Appearance(Who IN _1{UID}){What}),Actor(UID IN CoStar(Who IN _1{UID}){WhoElse})

Assorted examples

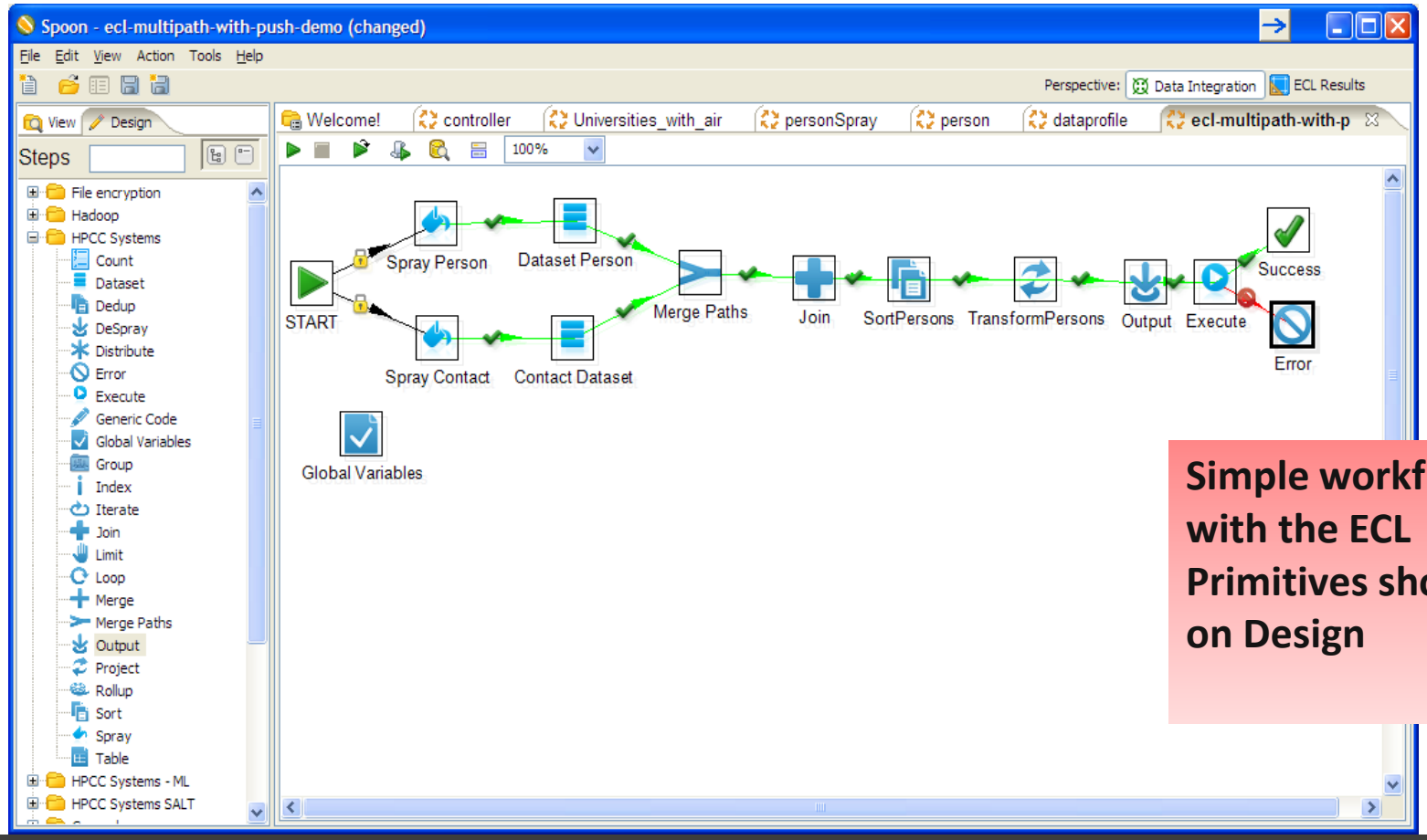
- Data integration using HPCC Flow
- Record linkage, entity disambiguation, entity resolution using SALT
- Data Visualization
- Exploratory data analysis
- Recommendation systems
- Search and retrieval
- Full text classifiers
- Scoring/statistical models
- Attribute manager
- Sentiment Analysis example
- Graph traversal problem handling (IMDB/Kevin Bacon example)
- Smart View
- SQL/JDBC integration
- Data Streaming/Kafka
- R integration
- Java integration
- Python integration

Data Integration

Data Integration using HPCC Flow

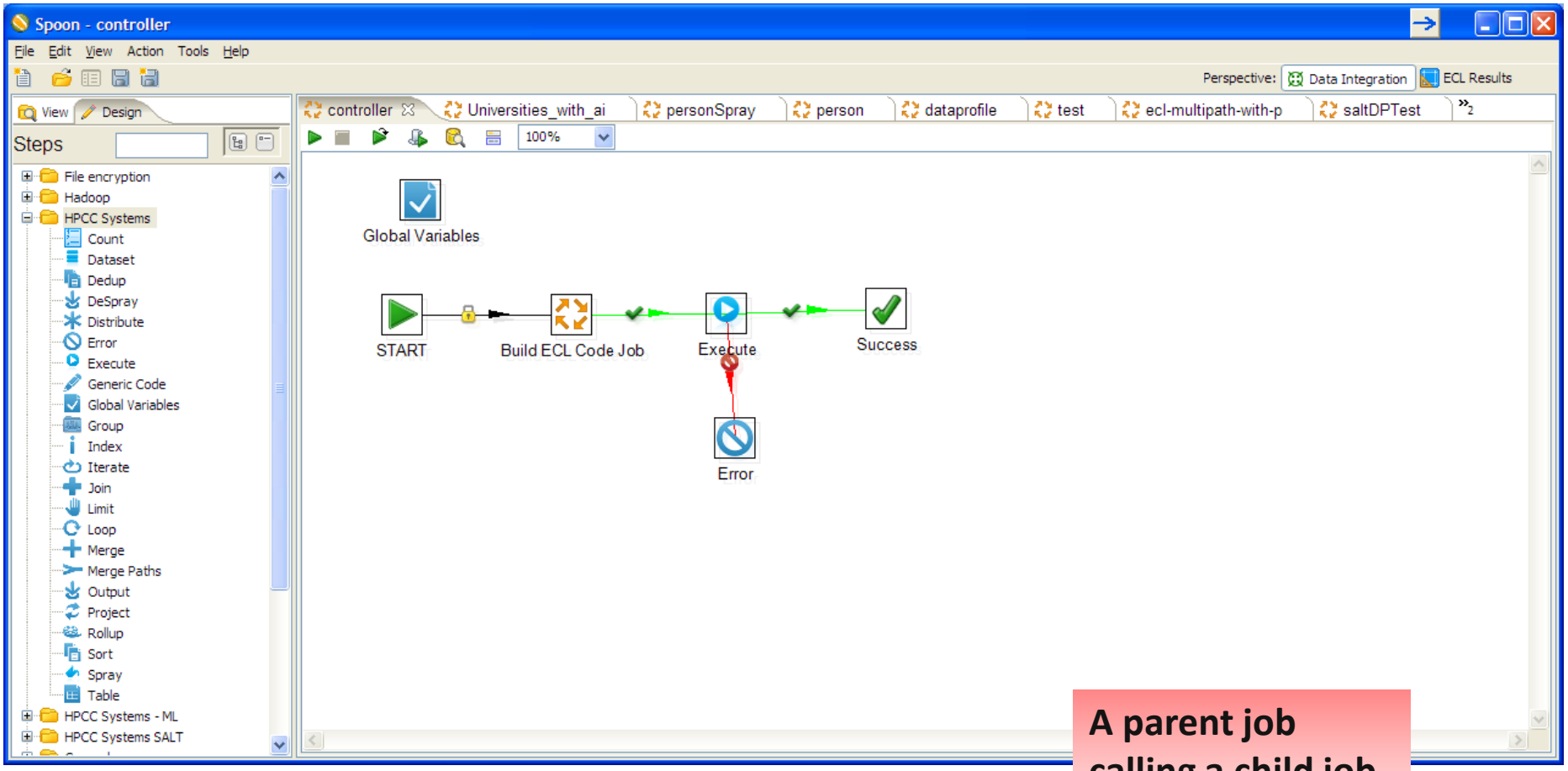
Drag and drop interface to data workflows on HPCC

Generates ECL code so it can be used also as a learning tool



Simple workflow, with the ECL Primitives shown on Design

Data Integration using HPCC Flow



A parent job calling a child job

Data Integration using HPCC Flow

Spoon - Universities_with_airplanes_subjob

File Edit View Action Tools Help

Perspective: Data Integration | ECL Results

View | Design

Steps

- File encryption
- Hadoop
- HPCC Systems
  - Count
  - Dataset
  - Dedup
  - DeSpray
  - Distribute
  - Error
  - Execute
  - Generic Code
  - Global Variables
  - Group
  - Index
  - Iterate
  - Join
  - Limit
  - Loop
  - Merge
  - Merge Paths
  - Output
  - Project
  - Rollup
  - Sort
  - Spray
  - Table
- HPCC Systems - ML
- HPCC Systems SALT

controller | Universities_with_ai | personSpray | person | dataprofile | test | ecl-multipath-with-p | saltDPTest

100%

Global Variables

AircraftMasterDataset

MasterAircraftProject

AircraftReferenceDataset

START

EngineReferenceDataset

Merge Paths

JoinUniversitiesWithAirplanes

JoinMasterReferenceEngine

JoinMasterReference

Universities With Airplanes Output

Success

UniversityDataset

UniversityProject

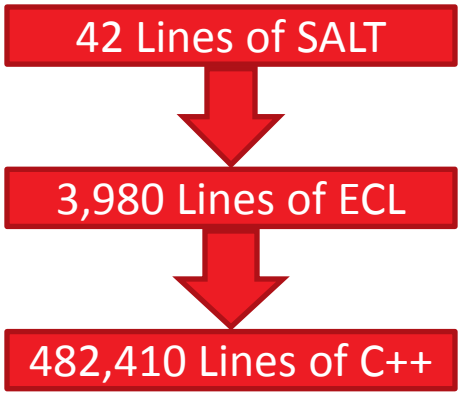
Child job called by a parent job

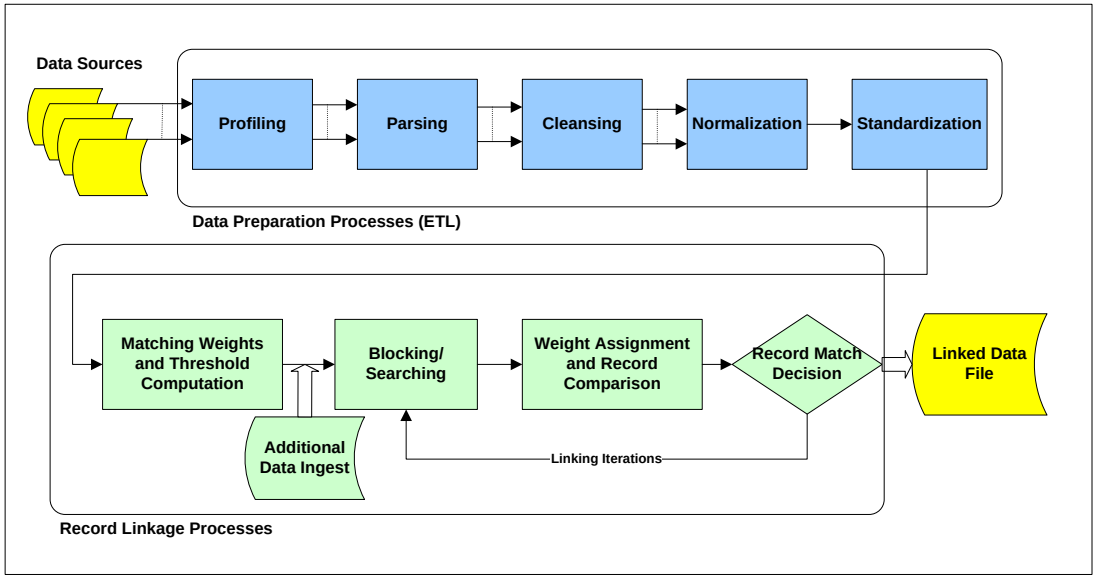
Get the source code: https://github.com/hpcc-systems/spoon-plugins

Record linkage, entity disambiguation, entity resolution

Scalable Automated Linking Technology (SALT)

- The acronym stands for “Scalable Automated Linking Technology”
- Template based ECL code generator
- Provides for automated data profiling, QA/QC, parsing, cleansing, normalization and standardization
- Sophisticated specificity based linking and clustering
- Significant productivity boost!





Rules based vs. probabilistic record linkage

Thoroughly employing statistical algorithms, SALT has the two significant advantages over rules:

1. SALT sufficiently explore all credible matches

INPUT

- Flavio Villanustre, Atlanta — Record 1
- Javio Villanustre, Atlanta — Record 2

SALT

Match, because the system has learnt that "Villanustre" is specific because the frequency of occurrence is small and there is only one present in Atlanta

Error

RULES

NO MATCH, because the rules determine that "Flavio" and "Javio" are not the same

2. SALT effectively minimizes false matches

INPUT

- John Smith, Atlanta
- John Smith, Atlanta

SALT

NO Match, because the system has learnt that "John Smith" is **not** specific because the frequency of occurrence is large and there are many present in Atlanta

Error

RULES

MATCH, because the rules determine that "John Smith" and the city for both the records match

Record Linkage: SALT – Scalable Automated Linking Technology

Another purpose of SALT is to perform record linking

Data Visualization

Data Visualization: Excel Integration

Access a Roxie query to display a dynamic data dashboard from within Excel:

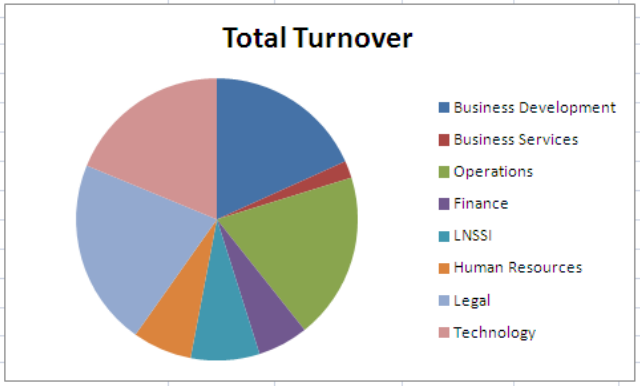
ID:

Begin Date: 2013/01/01

End Date:

Query

	department	total_employees	employees	turnover
1	Business Development	2229	24	1.08%
2	Business Services	240	27	11.25%
3	Operations	2332	38	1.63%
4	Finance	709	21	2.96%
5	LNSSI	965	19	1.97%
6	Human Resources	832	1	0.12%
7	Legal	2616	24	0.92%
8	Technology	2292	38	1.66%
9	Insurance	2568	21	0.82%

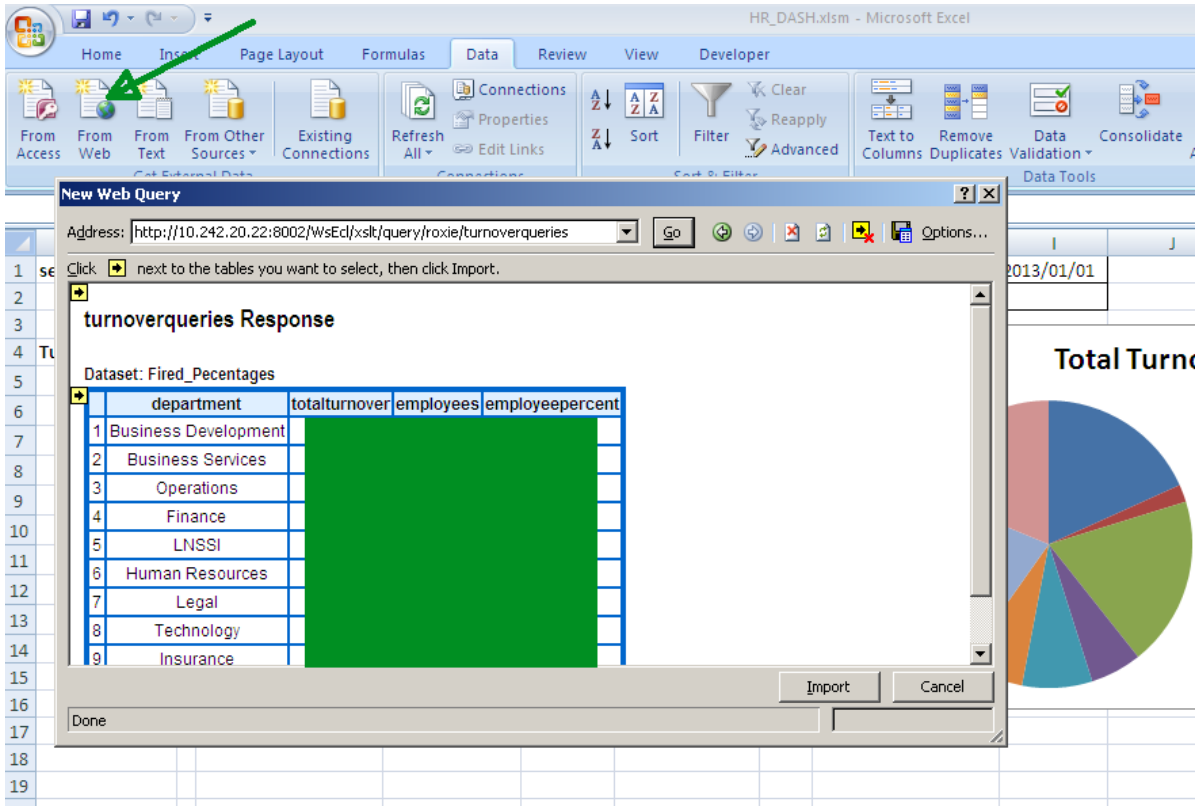


The Excel Visual Basic code to retrieve data:

```
Function refreshdata()
   Application.EnableEvents = False
   [Turnover].QueryTable.Connection = "URL;http://10.242.99.99:8002/WsEcl/xslt/query/roxie/excelqueries? asofdate=" + [qrydatecell].Text
   Call [Turnover].QueryTable.Refresh
   Application.EnableEvents = True
End Function
```

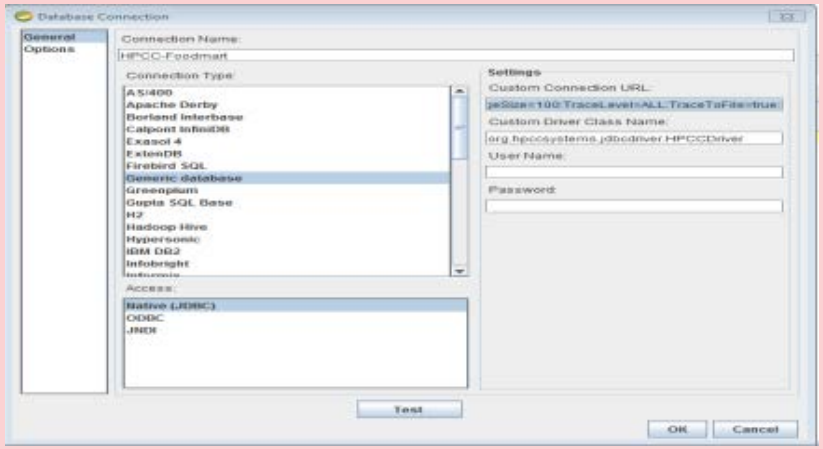
Data Visualization: Excel Integration…

Selecting the published Roxie Query in Excel:



Data Visualization: Pentaho Report Designer

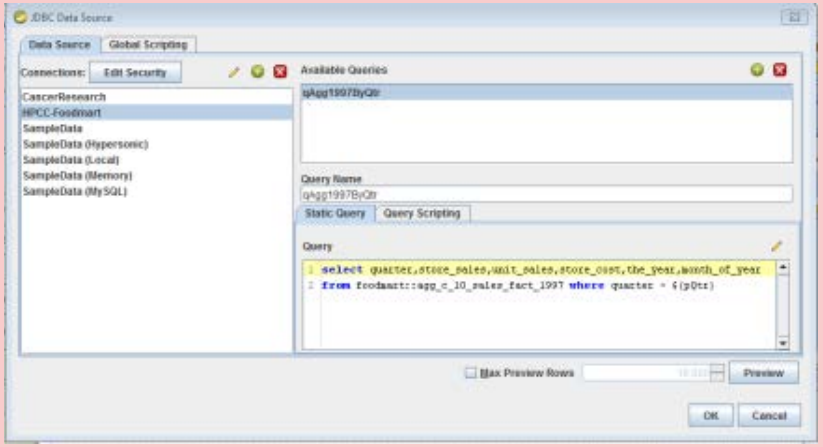
Desktop visual report designer
Reports can be saved locally or published to a **Pentaho BI Server**



Copy the HPCC JDBC Driver in directory \report-designer\lib\jdbc and specify the following values:

Connection Type: Generic Database
Custom Connection URL:
jdbc:hpcc:ServerAddress=99.999.999.9;Cluster=default:WsECLDirectPort=8008:EclResultLimit=100:QuerySet=thor:LazyLoad=true:PageSize=100:TraceLevel=ALL:TraceToFile=true:

Custom Driver Class Name:
org.hpccsystems.jdbcdriver.HPCCDriver



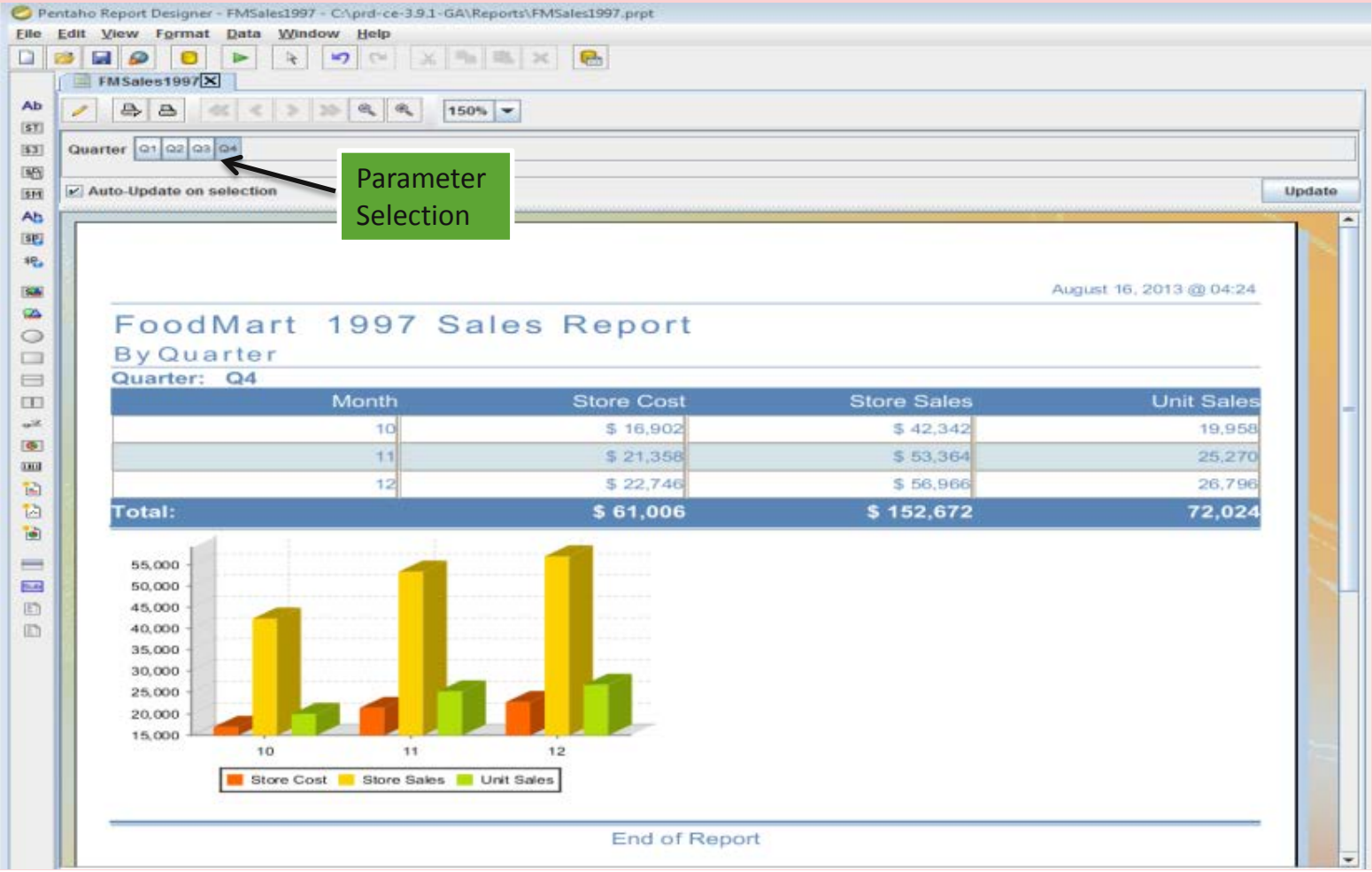
Reports/Charts can be created by writing a custom query or by calling a Roxie query.

Adhoc SQL Query:
select quarter, store_sales, unit_sales, store_cost,the_year, month_of_year
from foodmart::agg_c_10_sales_fact_1997 where quarter = ${pQtr}

Parameter

Roxie Query:
call roxie::all_cancers_for_state(GA)

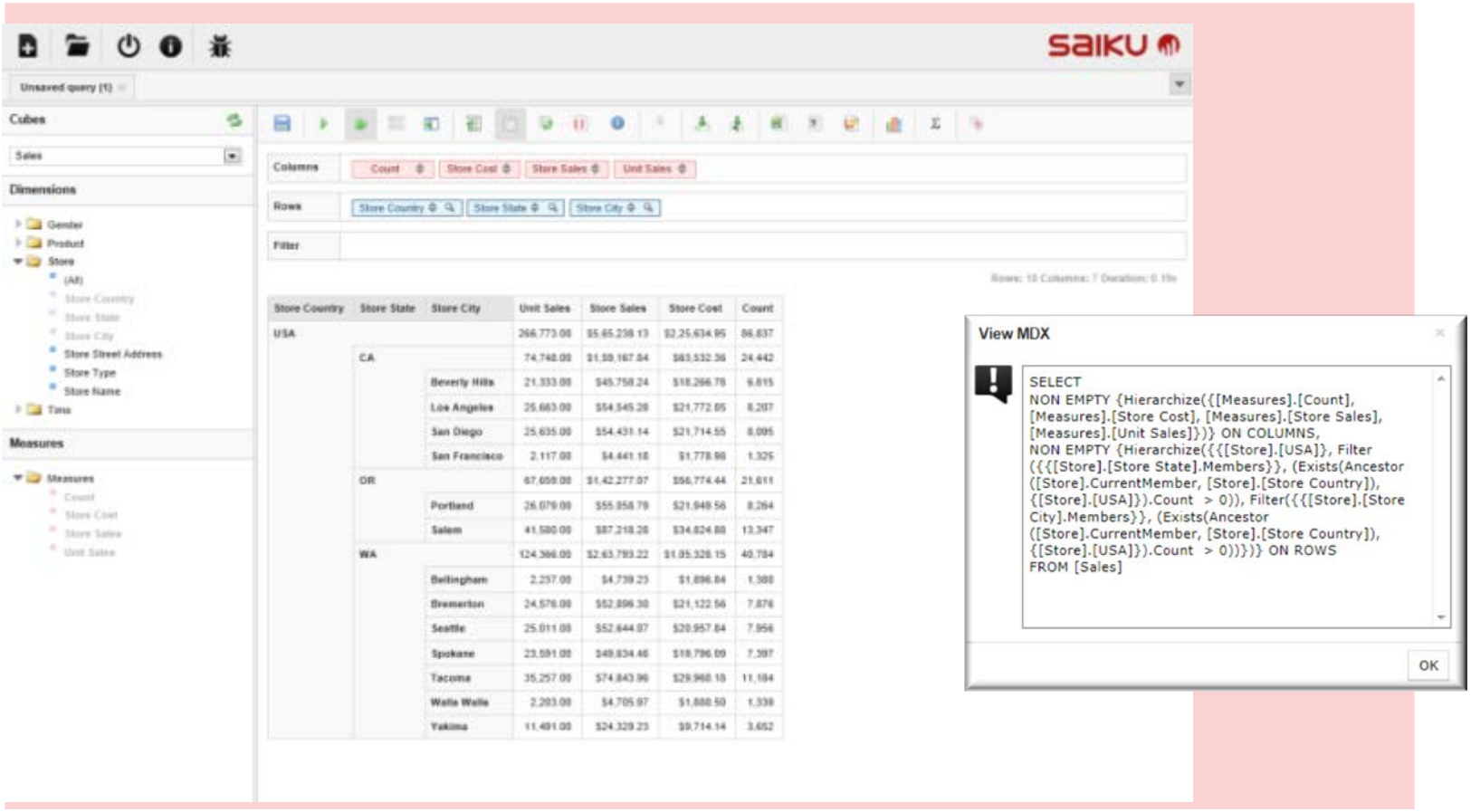
Data Visualization: Pentaho Report Designer



Download: http://reporting.pentaho.com/report_designer.php

Data Visualization: Mondrian using Saiku UI

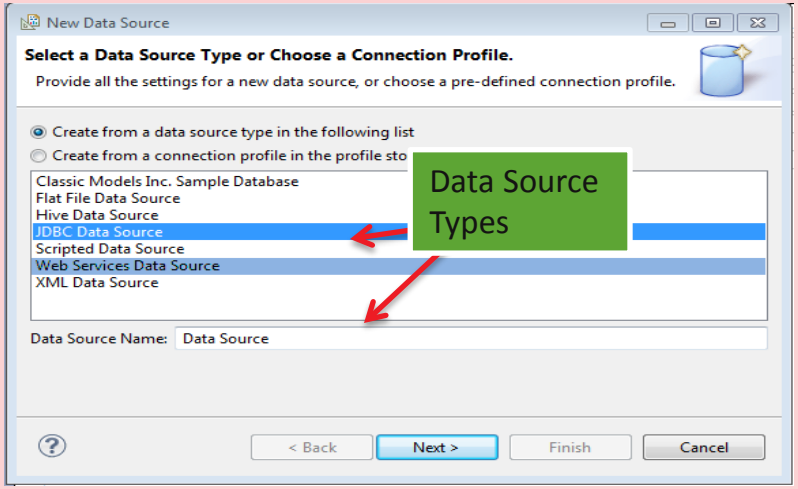
Real-time graphical online Analytical Processing (OLAP) on HPCC
Users explore business data by drilling into and cross-tabulating information with speed-of-thought response times to complex analytical queries.

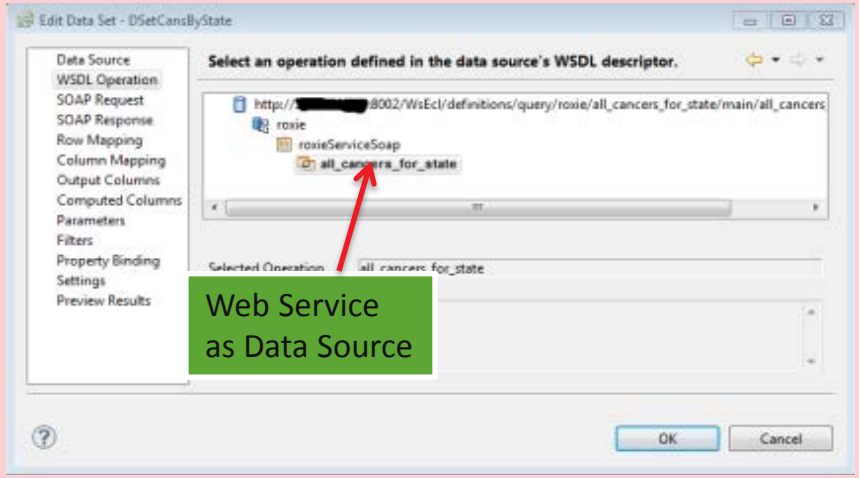


Download: http://meteorite.bi/saiku

Data Visualization: BIRT

Eclipse-based open source reporting system with HPCC

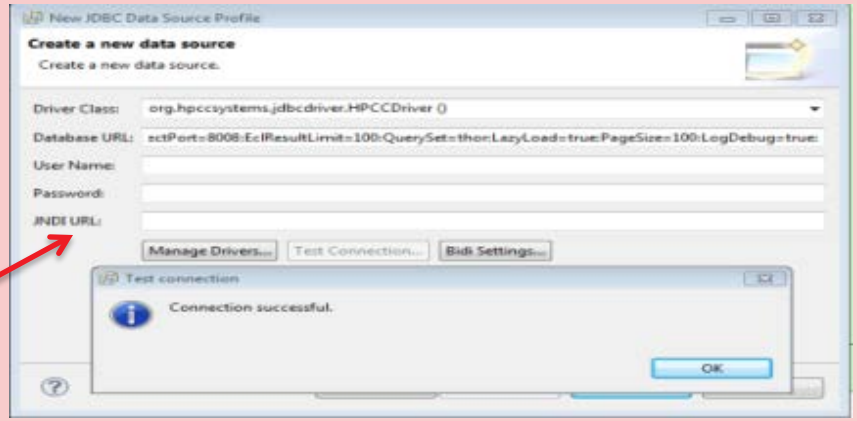




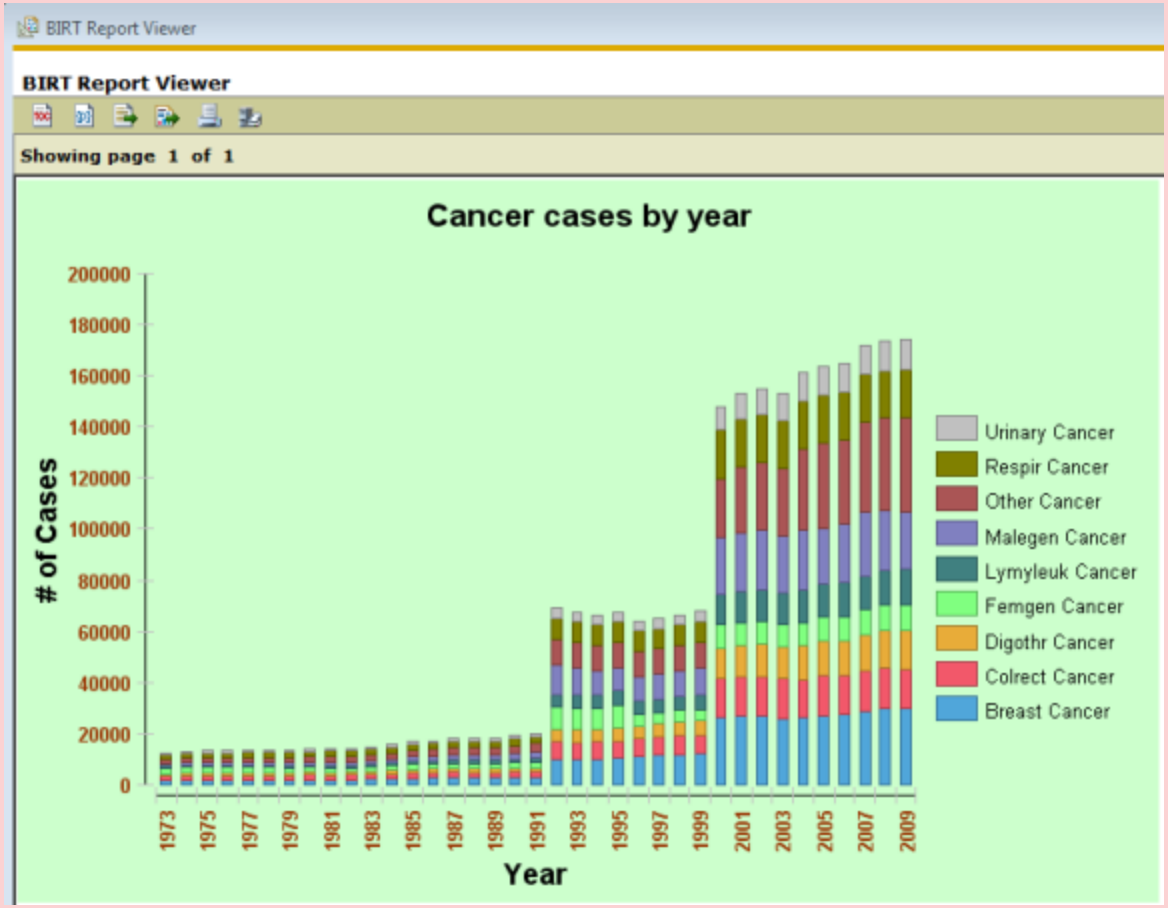
BIRT Data Source Types:

- JDBC (for AdHoc queries or Roxie queries)
- Web Service (Roxie queries)

JDBC Data Source



Data Visualization: BIRT Example Output



Download: http://download.eclipse.org/birt/downloads/

Data Visualization: Result Cell Formatting

Visualizations can be used to:

- Understand new data
- Demonstrate data/products
- Summarize data.

In the ECL Watch Tech Preview, users can specify specific output columns to contain “HTML” or “JavaScript”, rather than the default plain text. See example ECL code snippet below.

This allows the user to perform some “Cell Formatting” of the results, such as:

- Add textual formatting, e.g.: Text (bold text)
- Cell color: Make a cell Red/Yellow to alert the viewer that certain rows are important.
- Embed Hyperlinks: Embed a link to a preformatted Google search, or a link to a Google map with lat. and long. etc.

```
import SampleData.Cars;
import Bundles.d3;

r := record
   dataset(Cars.CarsRecord) cars;
   varstring pc__javascript;
end;

d3Chart := d3.Chart('cars', 'name', 'unused');
d := dataset([{Cars.CarsDataset, d3Chart.ParallelCoordinates}], r);
d;
```

Name the column with a trailing “__html” or “__javascript”

Data Visualization: Result Cell Formatting Output

HPCC Platform

192.168.1.201:8010/esp/files/stub.htm?Widget=HPCCPlatformWidget#

HPCC Platform

Wuid, User, More...

LOGGED IN AS:

Workunits Playground

Workunits | W20130813-052410 | W20130813-052421 | W20130813-052429 | W20130813-052434

W20130813-052434 | Variables | Outputs (1) | Inputs | Timers (38) | Graphs (1) | ECL | Helpers (3) | Playground | XML

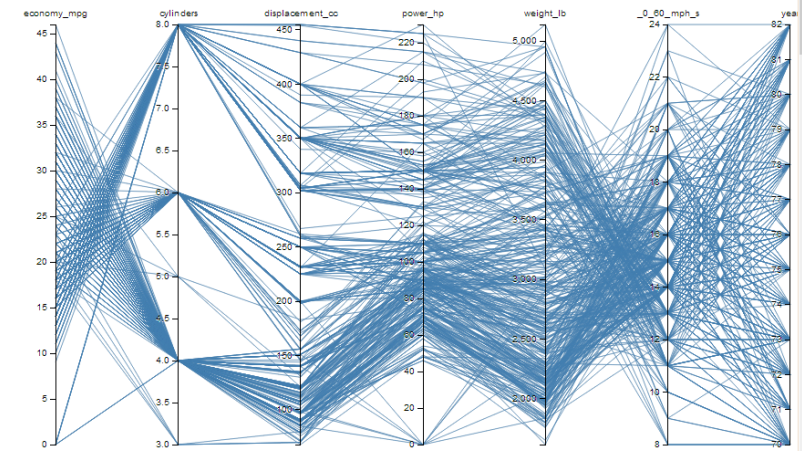
Outputs | Result 1

Download: Zip GZip XLS

##	cars							
1	name	economy_mpg	cylinders	displacement_cc	power_hp	weight_lb	_0_60_mph_s	year
	AMC Ambassador Brougham	13	8	360	175	3821	11	73
	AMC Ambassador DPL	15	8	390	190	3850	8	70
	AMC Ambassador SST	17	8	304	150	3672	11	72
	AMC Concord DL 6	20	6	232	90	3265	18	79
	AMC Concord DL	18	6	258	120	3410	15	78
	AMC Concord DL	23	4	151	0	3035	20	82
	AMC Concord	19	6	232	90	3210	17	78
	AMC Concord	24	4	151	90	3003	20	80
	AMC Gremlin	18	6	232	100	2789	15	73
	AMC Gremlin	19	6	232	100	2634	13	71
	AMC Gremlin	20	6	232	100	2914	16	75
	AMC Gremlin	21	6	199	90	2648	15	70
	AMC Hornet Sportabout (Wagon)	18	6	258	110	2962	13	71
	AMC Hornet	18	6	199	97	2774	15	70
	AMC Hornet	18	6	232	100	2945	16	73
	AMC Hornet	19	6	232	100	2901	16	74
	AMC Hornet	22	6	232	90	3085	17	76
	AMC Matador (Wagon)	14	8	304	150	4257	15	74
	AMC Matador (Wagon)	15	8	304	150	3892	12	72
	AMC Matador	14	8	304	150	3672	11	73
	AMC Matador	15	6	258	110	3730	19	75
	AMC Matador	15	8	304	120	3962	13	76
	AMC Matador	16	6	258	110	3632	18	74
	AMC Matador	18	6	232	100	3288	15	71
	AMC Pacer D/L	17	6	258	95	3193	17	76
	AMC Pacer	19	6	232	90	3211	17	75
	AMC Rebel SST (Wagon)	0	8	360	175	3850	11	70
	AMC Rebel SST	16	8	304	150	3433	12	70
	AMC Spirit DL	27	4	121	80	2670	15	79
	Audi 100 LS	20	4	114	91	2582	14	73
	Audi 100 LS	23	4	115	95	2694	15	75
	Audi 100 LS	24	4	107	90	2430	14	70
	Audi 4000	34	4	97	78	2188	15	80
	Audi 5000	20	5	131	103	2830	15	78
	Audi 5000S (Diesel)	36	5	121	67	2950	19	80
	Audi Fox	29	4	98	83	2219	16	74
	BMW 2002	26	4	121	113	2234	12	70
	BMW 320i	21	4	121	110	2600	12	77
	Buick Century 350	13	8	350	175	4100	13	73

1 - 1 of 1 results

pc__javascript



Data Visualization: Result Cell Formatting – Basic Chart

HPCC Platform

192.168.1.201:8010/esp/files/stub.htm?Widget=HPCCPlatformWidget#/stub/ECL/Workunits/W20130813-052410

Workunits | Playground

Workunits | W20130813-052410 | W20130813-052421 | W20130813-052429 | W20130813-052434

W20130813-052410 | Variables | Outputs (1) | Inputs (1) | Timers (38) | Graphs (1) | ECL | Helpers (3) | Playground | XML

Outputs | Result 1

Download: Zip GZip XLS

state	numrows	_data	bar_javascript	pie_javascript	bubble_javascript

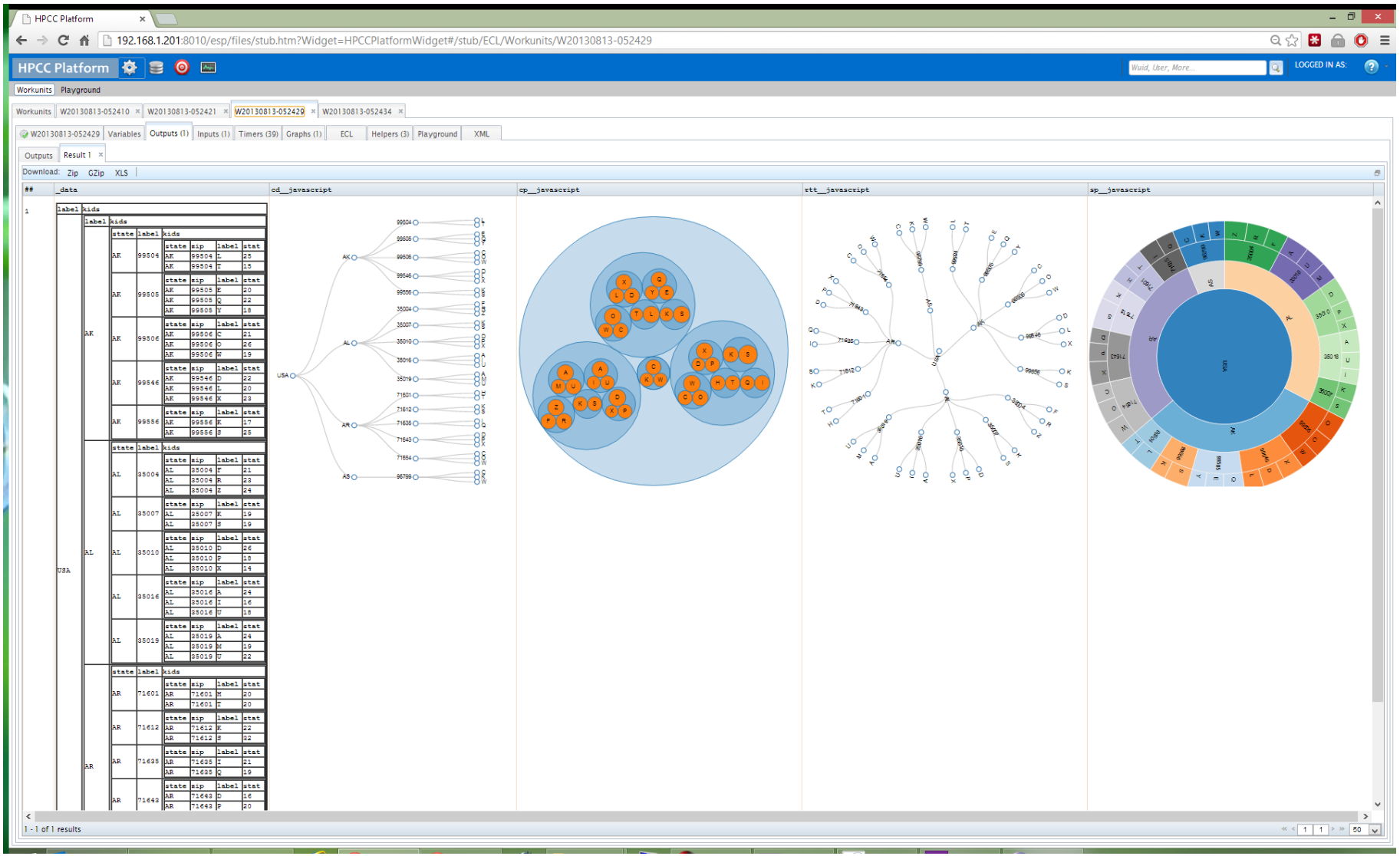
AK	P	66
AK	Q	102
AK	R	17
AK	S	231
AK	T	97
AK	U	67
AK	V	25
AK	W	72
AK	X	88
AK	Y	151
AK	Z	69

AL 26

state	middleinitial	stat
AL	A	501
AL	B	610
AL	C	365
AL	D	334
AL	E	348
AL	F	325
AL	G	306
AL	H	339
AL	I	315
AL	J	579
AL	K	404
AL	L	426
AL	M	533
AL	N	377
AL	O	316
AL	P	346
AL	Q	264
AL	R	451
AL	S	252
AL	T	402
AL	U	493
AL	V	592
AL	W	357
AL	X	357
AL	Y	288
AL	Z	308

1 - 50 of 56 results

Data Visualization: Result Cell Formatting – Tree Structures



Data Visualization: Result Cell Formatting – Graphs

HPCC Platform

192.168.1.201:8010/esp/files/stub.htm?Widget=HPCCPlatformWidget#/stub/ECL/Workunits/W20130813-052421

Workunits | Playground

Workunits | W20130813-052410 | W20130813-052421 | W20130813-052429 | W20130813-052434

W20130813-052421 | Variables | Outputs (1) | Inputs | Timers (38) | Graphs (1) | ECL | Helpers (3) | Playground | XML

Outputs | Result 1

Download: Zip GZip XLS

1

vertices

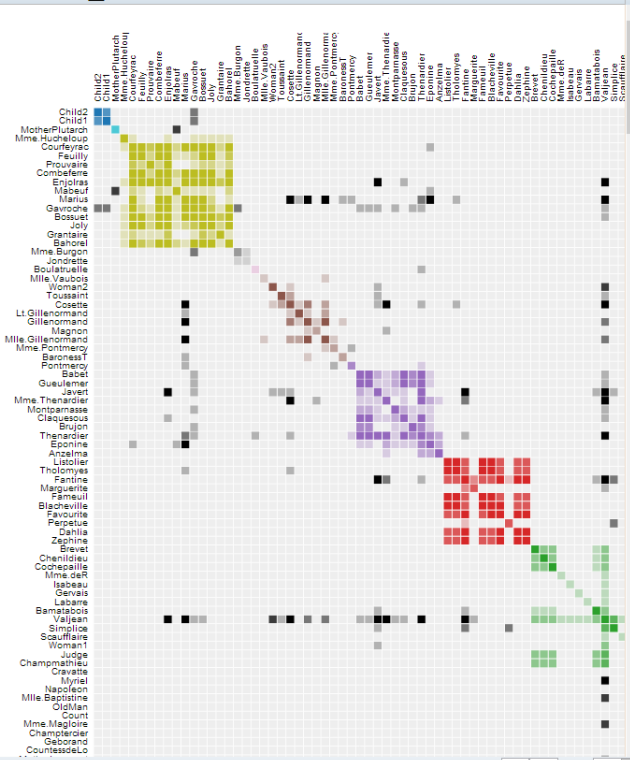
id	name	category
0	Myriel	1
1	Napoleon	1
2	Mlle.Baptistine	1
3	Mme.Magloire	1
4	CountessdeLo	1
5	Geborand	1
6	Champtercier	1
7	Cravatte	1
8	Count	1
9	OldMan	1
10	Labarre	2
11	Valjean	2
12	Marguerite	3
13	Mme.deR	2
14	Isabeau	2
15	Gervais	2
16	Tholomyes	3
17	Listolier	3
18	Fameuil	3
19	Blacheville	3
20	Favourite	3
21	Dahlia	3
22	Zephine	3
23	Fantine	3
24	Mme.Thenardier	4
25	Thenardier	4
26	Cosette	5
27	Javert	4
28	Fauchelevent	0
29	Bamatabois	2
30	Perpetue	3
31	Simplice	2
32	Scaufflaire	2
33	Woman1	2
34	Judge	2
35	Champmathieu	2
36	Brevet	2
37	Chenildieu	2
38	Cochepaille	2

edges

source	target	weight
1	0	1
2	0	8
3	0	10
3	2	6
4	0	1
5	0	1
6	0	1
7	0	1
8	0	2
9	0	1
11	10	1
11	3	3
11	2	3
11	0	5
12	11	1
13	11	1
14	11	1
15	11	1
17	16	4
18	16	4
18	17	4
19	16	4
19	17	4
19	18	4
20	16	3
20	17	3
20	18	3
20	19	4
21	16	3
21	17	3
21	18	3
21	19	3
21	20	5
22	16	3
22	17	3
22	18	3
22	19	3
22	20	4
22	21	4

fd_javascript

cooccurrence_javascript



1 - 1 of 1 results

Exploratory Data Analysis

Exploratory Data Analysis

The EDA tool set will be tools added to HPCC Flow that allow analytics on big data. The EDA tools will not provide all the functionality of existing tools so an additional benefit is the ability to reduce a "Big Data" problem down to a statistically significant subset of random records to be used in existing analytic tools such as R and SAS.

EDA Procedures

- Frequency
- Mean, Median, Mode, Standard Deviation, Percentiles, Min, Max (Essentially Proc Univariate)
- Rank descending and ascending
- Tabulate procedure
- Random number generators
- Simple math ops such as x/y for each observation
- Options to easily add columns of data created by doing math on one or more existing columns of data
- Correlation statistics between two variables

EDA Visualization

- Scatter Plots
- Line Plots
- Histograms
- Pie charts
- GUI will allow user to define which data is plotted against the X and Y axis
- Graph options will include Line, Bar, Pie and Histogram

Sentiment Analysis

Sentiment Analysis: ECL-ML

HPCC Systems and its ECL-ML library includes an extensible set of fully parallel Machine Learning (ML) and Matrix processing algorithms to assist with business intelligence; covering supervised and unsupervised learning, document and text analysis, statistics and probabilities, and general inductive inference related problems.

1. Download the ML Library http://hpccsystems.com/ml
2. Extract the contents of the zip file to the ECL IDE source folder.
3. Reference the library in your ECL source using a import statement as shown in the example:

```
IMPORT * FROM ML;
IMPORT * FROM ML.Cluster;
IMPORT * FROM ML.Types;

x2 := DATASET([ {1, 1, 1}, {1, 2, 5}, {2, 1, 5}, {2, 2, 7},
{3, 1, 8}, {3, 2, 1}, {4, 1, 0}, {4, 2, 0}, {5, 1, 9}, {5, 2,
3}, {6, 1, 1}, {6, 2, 4}, {7, 1, 9}, {7, 2,
4}],NumericField);

c := DATASET([ {1, 1, 1}, {1, 2, 5}, {2, 1, 5}, {2, 2, 7},
{3, 1, 9}, {3, 2, 4}],NumericField);

x3 := Kmeans(x2,c);

OUTPUT(x3);
```

ECL-ML Classification: Naïve Bayes Classification

ML.Classify tackles the problem: "given I know these facts about an object; can I predict some other value or attribute of that object." For example, can I predict whether sentiment of a tweet is positive or negative?

Using a classifier in ML involves three logical steps:

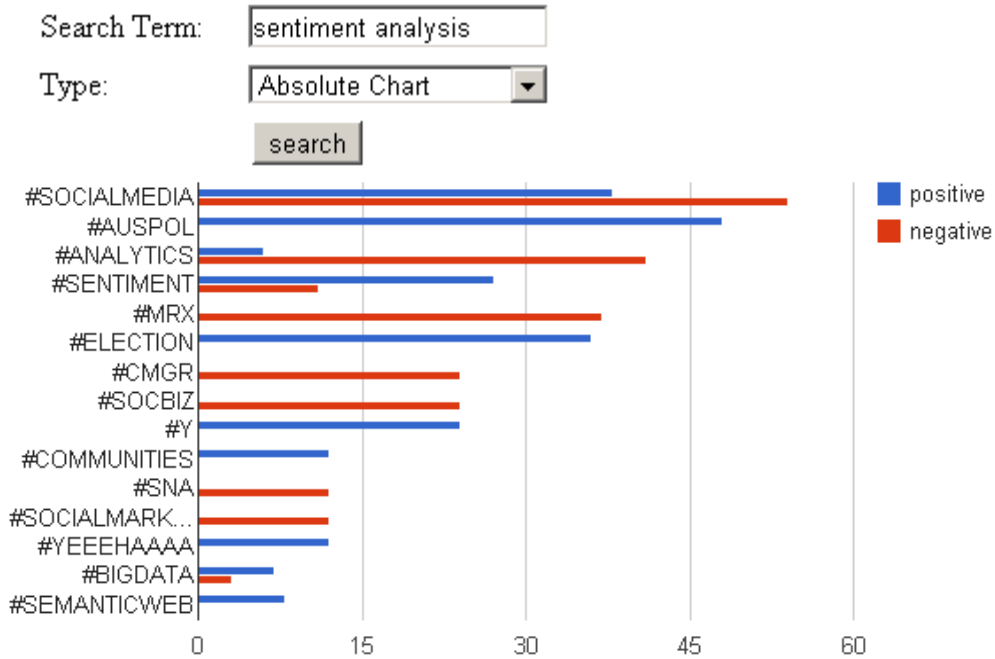
1. Learning the model from a training set of data that has been classified externally.
2. Testing. Getting measures of how well the classifier fits.
3. Classifying. Apply the classifier to new data in order to give it a classification

```
IMPORT ML;

/* Use ML.Docs module to pre-process tweet text and generate
classification training set: IndependentDS, ClassDS

dRaw  - collection of positive and negative sentiment tweets
dTokens := ML.Docs.Tokenize.Split(ML.Docs.Tokenize.Clean(dRaw));
dLexicon := ML.Docs.Tokenize.Lexicon(dTokens);
ML.Docs.Trans(ML.Docs.Tokenize.ToO(dTokens,dLexicon)).WordsCounted;
*/
BayesModule := ML.Classify.NaiveBayes;
// Learning the model
Model := BayesModule.LearnD(IndependentDS, ClassDS);
// Testing
TestModule := BayesModule.TestD(IndependentDS, ClassDS);
OUTPUT(TestModule);
//Classifying
Results := BayesModule.ClassifyD(IndependentDS, Model);
OUTPUT(Results);
```

Sentiment Analysis: “Sentilyze” Twitter Sentiment Classification

Sentilyze uses HPCC Systems and its ML-Library to classify tweets with positive or negative sentiment.

Enter a search query and receive real-time tweets via the Twitter API in real-time.



Try the demo:
http://hpccsystems.com/demos/twitter-sentiment
Get the source code:
https://github.com/hpcc-systems/ecl-ml

Example code of a dataset of tweets that are classified using both Keyword Count and Naïve Bayes sentiment classifiers:

```
IMPORT Sentilyze;

Tweets :=
DATASET('~SENTILYZE::TWEETS',Sentilyze.Types.TweetType.CSV);

rawTweets := Sentilyze.PreProcess.ConvertToRaw(Tweets);

processTweets := Sentilyze.PreProcess.RemoveAnalysis(rawTweets)

kcSentiment := Sentilyze.KeywordCount.Classify(processTweets);

nbSentiment := Sentilyze.NaiveBayes.Classify(processTweets);

OUTPUT(kcSentiment,NAMED('TwitterSentiment_KeywordCount'));

OUTPUT(nbSentiment,NAMED('TwitterSentiment_NaiveBayes'));
```

Graph Traversal Problem Handling

Graph Traversal Problem Handling

Data analysis techniques using HPCC Systems ECL can be applied to social graph traversal problems for finding customer insight and detection of important data patterns and trends.

Get the example code:
http://hpccsystems.com/download/docs/six-degrees

```
/* ******************************************************************************
ATTRIBUTE PURPOSE:
 Produce a series of sets for Actors and Movies that are : distance-0
 away (KBacons Direct movies ), distance-2 Away KBacon's Costars Movies ,
 distance-3 away - Movies of Costars of Costars etc all the way upto level 7

 The nested attributes below are shown here together for the benefit of the reader.

 Notes on variable naming convention used for costars and movies
 KBMovies                :  Movies  Kevin Bacon Worked in     (distance 0)
 KBCoStars               :  Stars who worked in KBMovies      (distance 1)
 KBCoStarMovies          :  Movies worked in by KBCoStars
                              except KBMovies   (distance 1)
 KBCo2Stars              :  Stars(Actors) who worked in KBCoStarMovies (distance 2)
 KBCo2StarMovies         :  Movies worked  in by KBCo2Stars
                              except KBCoStarMovies    (distance 2)
 KBCo3Stars              :  Stars(Actors) who worked in KBCo2StarMovies (distance 3)
 KBCo3StarMovies         :  Movies worked  in by KBCo3Stars
                              except KBCo2StarMovies   (distance 3)
etc..
****************************************************************************** */

IMPORT Std;
IMPORT IMDB;

EXPORT KevinBaconNumberSets := MODULE
  // Constructing a proper name match function is an art within itself
  // For simplicity we will define a name as matching if both first and last name
  //are found within the string

  NameMatch(string full_name, string fname,string lname) :=
    Std.Str.Find(full_name,fname,1) > 0 AND
    Std.Str.Find(full_name,lname,1) > 0;

  //------ Get KBacon Movies
  AllKBEntries := IMDB.ActorsInMovies(NameMatch(actor,'Kevin','Bacon'));
  EXPORT KBMovies := DEDUP(AllKBEntries, movie, ALL); // Each movie should ONLY occur once

  //------ Get KBacon CoStars
  CoStars := IMDB.ActorsInMovies(Movie IN SET(KBMovies,Movie));
  EXPORT KBCoStars := DEDUP( CoStars(actor<>'Kevin Bacon'), actor, ALL);
```

Getting Useful Information from Data using Links and Degrees of Separation aka "*Kevin Bacon example*"

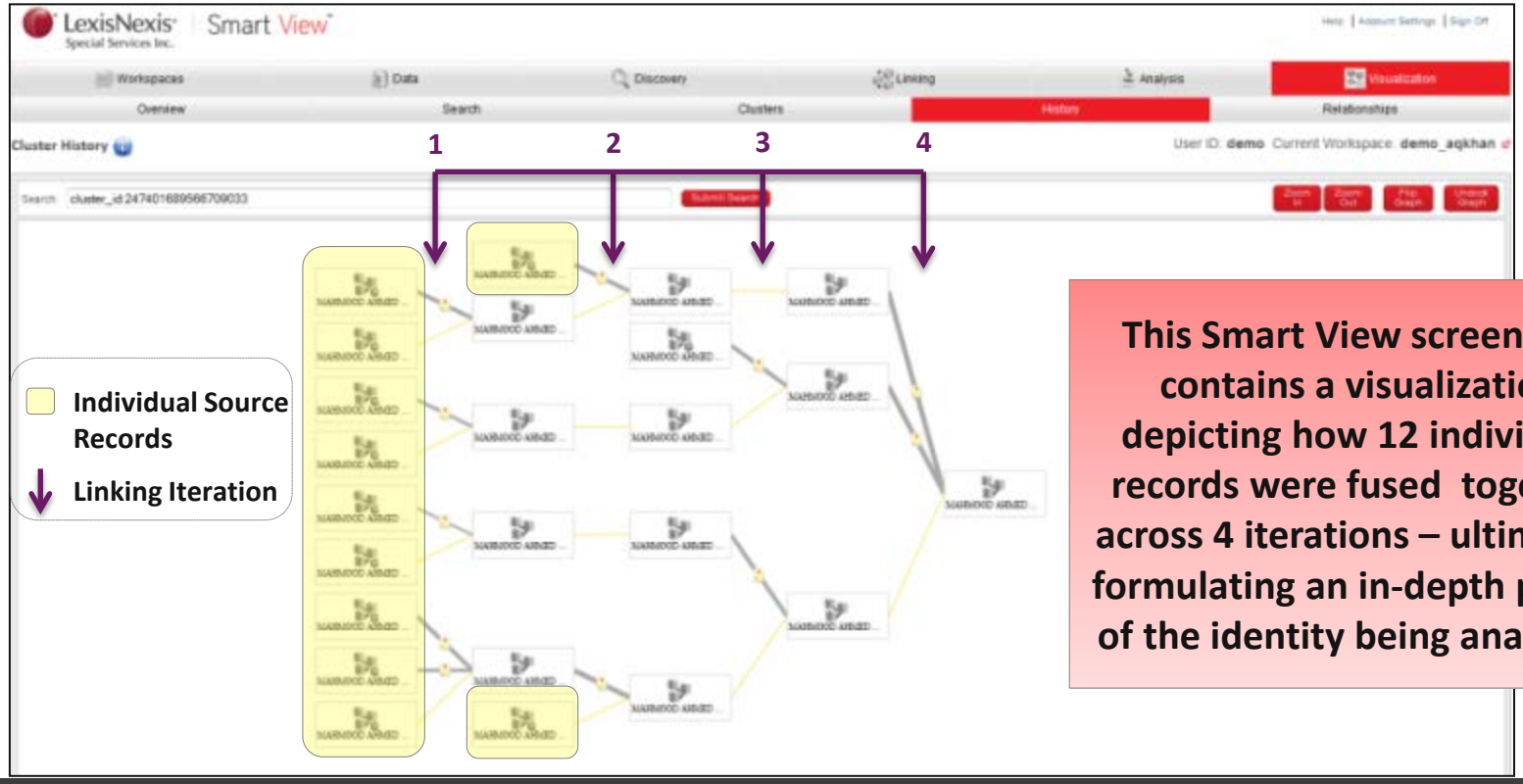
```
IMPORT IMDB;
IMDB.KevinBaconNumberSets.doCounts;
```

KB Movies	71
KB Co Stars	3520
KB Co Star Movies	33504
KB Co 2 Stars	430145
KB Co 2 Star Movies	251867
KB Co 3 Stars	896009
KB Co 3 Star Movies	51650
KB Co 4 Stars	102729
KB Co 4 Star Movies	2634
KB Co 5 Stars	6080
KB Co 5 Star Movies	190
KB Co 6 Stars	450
KB Co 6 Star Movies	14
KB Co 7 Stars	22

Smart View

Smart View™ Entity Resolution & Visualization

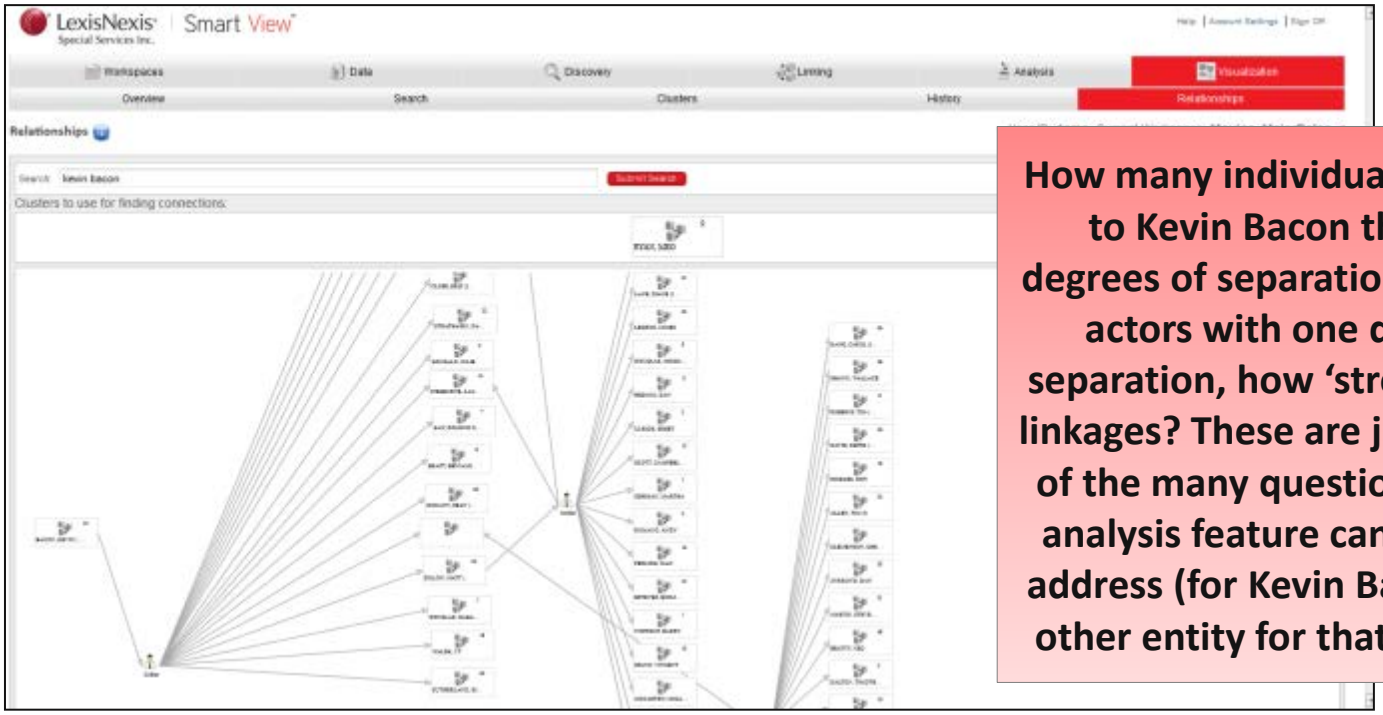
Smart View™ (SV) generates comprehensive profiles of 'entities' – people, places, and organizations – from a 'perfect storm' of big data (ie, disparate, complex and massive datasets requiring fusion for time and mission-critical applications). The combination of SV's intuitive user interface, along with its leveraging of a patented, statistically-based clustering technique, make it a compelling option for the data scientist or business analyst faced with complex entity resolution challenges.



This Smart View screenshot contains a visualization depicting how 12 individual records were fused together across 4 iterations – ultimately formulating an in-depth profile of the identity being analyzed.

Smart View™ Link Analysis & Visualization

The more comprehensive your entity profiles are, the greater the probability that non-obvious linkages between them will be uncovered. Smart View's link analysis feature allows the end-user to define a series of relationships, then uncover such relationships through an iterative, statistical process (similar to how SV's entity resolution problem is solved). To aide the data scientist in interpreting what their link analysis results signify, Smart View includes a set of relationship visualizations that display dynamically according to 'point' and 'click' commands.



How many individuals are linked to Kevin Bacon through 3 degrees of separation? For those actors with one degree of separation, how 'strong' are the linkages? These are just a sample of the many questions SV's link analysis feature can help users address (for Kevin Bacon, or any other entity for that matter ☺)

SQL/ JDBC Integration

SQL/JDBC Integration

The HPCC JDBC Driver provides SQL-based access to HPCC Systems data files and published queries.

This allows you to connect to the HPCC Systems platform through your favorite JDBC client and access your data without the need to write a single line of ECL!

- Analyze HPCC data using many JDBC Client User Interfaces
- Supports simple SQL SELECT or CALL syntax
  - Access HPCC data files as DB Tables
  - Access published queries as DB Stored Procedures
- Harnesses the full power of HPCC under the covers
  - Submitted SQL request generates ECL code which is submitted, compiled, and executed on your target cluster
  - Automatic Index fetching capabilities for quicker data fetches
- Creates entry-point for programmatic data access
- Leverage HPCC data and JDBC client functionality without need to learn and write ECL!
  - Opens the door for non ECL programmers to access HPCC data.

SQL/JDBC Integration: Example Code

```
public class HpccConnect {
    public static void main(String[] args)
    {
        Connection con=null;;
        try {

            //1. Initialize HPCC Connection
Class.forName("org.hpccsystems.jdbcdriver.HPCCDriver");
            con=DriverManager.getConnection("jdbc:hpcc:ServerAddress=99.99.99.99;
             Cluster=default:WsECLDirectPort=8010:EclResultLimit=100:QuerySet=thor:LazyLoad=true:PageSize=100:LogDebug=true:","","");

                //2. Write the Query to select foodMart::currency table
                PreparedStatement pst=con.prepareStatement("select conversion_ratio,currency_id,currency from foodmart::currency");
               pst.clearParameters();
               ResultSet rs=pst.executeQuery();

                //3. Print the headers to the console
               System.out.println("conversion_ratio\t\tcurrency_id\t\tcurrency");

                //4. Loop through all the records and write the information to the console
                while(rs.next()) {
                    System.out.println(rs.getObject(1)+"\t\t"+rs.getString(2)+"\t\t"+rs.getString(3));
               }

                //5. Close the connection
               con.close();

                }
        catch(Exception e){
                    e.printStackTrace();
               }
        }
    }
```

Download JDBC Driver:
http://hpccsystems.com/products-and-services/products/plugins/JDBC-Driver

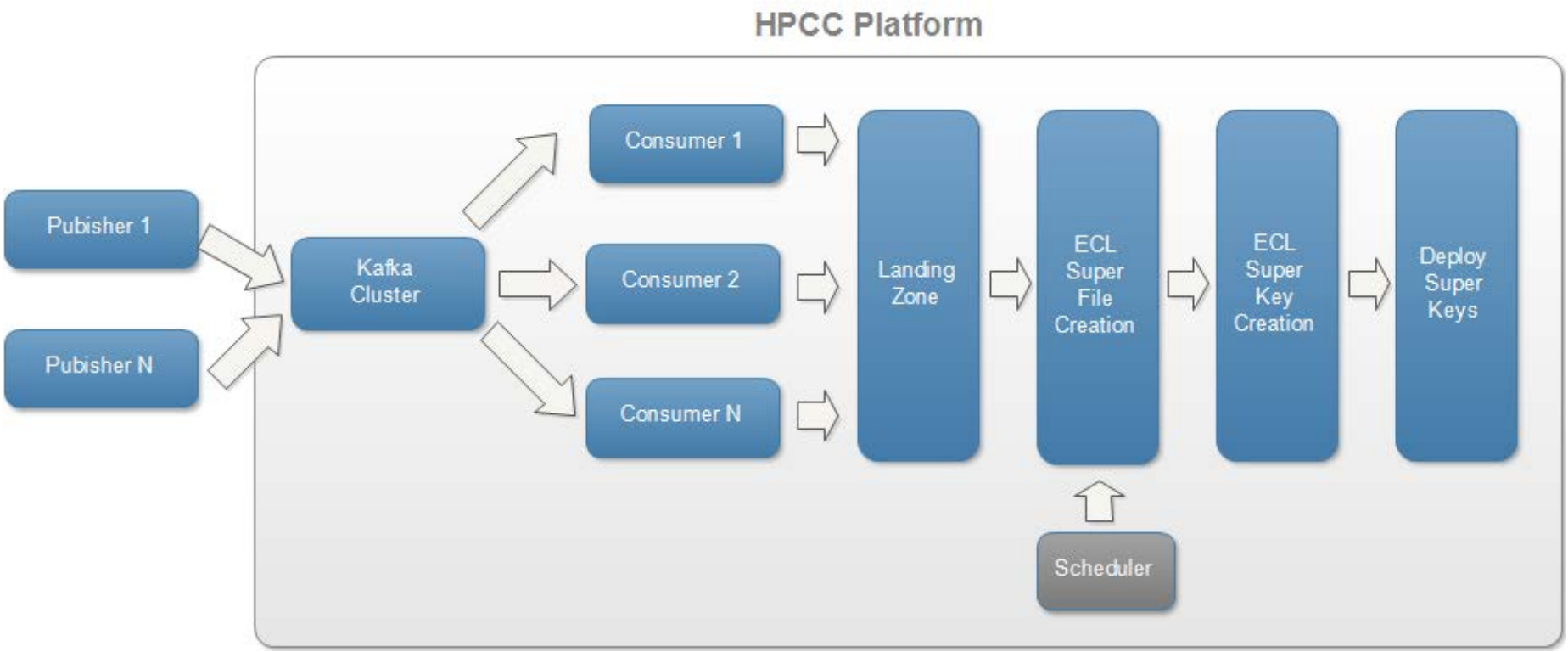
Data Streaming

Data Streaming: Apache Kafka

Apache Kafka is a distributed publish-subscribe messaging system. It is designed to support the following:

- Persistent messaging with O(1) disk structures that provide constant time performance even with many TB of stored messages.
- High-throughput: even with very modest hardware Kafka can support hundreds of thousands of messages per second.
- Explicit support for partitioning messages over Kafka servers and distributing consumption over a cluster of consumer machines while maintaining per- partition ordering semantics.
- Horizontally Scalable

Data Streaming: How we are using Apache Kafka



R Integration

R Integration: rHPCC Plugin

rHPCC is an R package providing an interface between R and HPCC Systems.

```
# Creates an ECL "PROJECT" definition.
# The PROJECT function processes through all records in the recordset performing
#  the transform function on each record in turn.
# Example ECL CODE: PROJECT(recordset,TRANSFORM(record,SELF := LEFT));

ECLProject <- setRefClass("ECLProject", contains="ECLDataset", fields = list(name = "character", inDataset="ECLDataset",
  outECLRecord="ECLRecord", def = "character"),
                    methods = list(
                     getName = function() {
                      result <- name
                     },

                     addField = function(id, value) {
                      def <<- paste(def, "  ", id, " := ", value, ";")
                     },

                     print = function() {
                      result <- paste (name, " := PROJECT( ", inDataset$getName(), ", TRANSFORM(", outECLRecord$getName() ,
  ",",def, " ));")
                     }
                     )
          )
```

R Integration: Code Example

ECL Code Snippet using EMBED(R)

```
IMPORT R;

EXAMPLE 1:

SET OF INTEGER2 sortArray ( SET OF INTEGER2 val) := EMBED(R)
    sort(val);    // R code here
ENDEMBED;

arr := sortArray ([-111, 0, 113, 12, 45, 5]);
arr[3];   // Returns 5

EXAMPLE 2:

STRING cat(VARSTRING what, VARSTRING who) := EMBED(R)
  paste(what,who)     // R code here
ENDEMBED;

result := cat('Hello', 'World');
result;   // Hello World
```

Java Integration

Java Integration: Java ECL API

Java ECL API was developed along side the HPCC Flow tool building the necessary JAVA code base to handle generating the ECL code, calling the Cluster via the SOAP interface, and retrieving results and server status updates. A secondary goal to this project is to provide examples on using the HPCC Systems Platform from another programming language in general.

Compile and submit to cluster

Java ECL API Example with no client side compile check

```
String eclCode = “output(‘Hello World’);”;
//getECLSoap() is a class used that sets up the server connection and returns a ECLSoap class
ECLSoap es = getECLSoap();
boolean isValid = false;
ArrayList dsList = null;
String outStr = "";

isValid = es.executeECL(eclCode);
String wuid = es.getWuid();
if(!this.isValid){
        this.error += "\r\nFailed to execute code on the cluster, please verify your settings\r\n";
 }
```

To check the ECL code with a local compiler call:

```
boolean isError = false;
boolean isWarning = false;
String errorMessage = (es.syntaxCheck(eclCode)).trim();
if(es.getErrorCount() > 0){
        isError = true;
}
 if(es.getWarningCount() > 0){
        isWarning = true;
}
```

Java Integration: Java ECL API – Code Generators

ECL Primitive

- Count
- Dataset
- Dedup
- DeSpray
- Distribute
- FromField
- Group
- Index
- Iterate
- Join
- Limit
- Loop
- Merge
- Output
- Project
- Rollup
- Sort
- Spray
- Table
- ToField

SALT

- Data Profile Report
- Concepts
- Specificities
- Hygiene Report
- Internal Linking

ML Library

- Associate
- Build Classify
- Classification
- Classify
- Discretize
- KMeanes
- Naïve Bayes

Java Integration: Java ECL API – ECL Primitive Code Generator

Render ECL Example

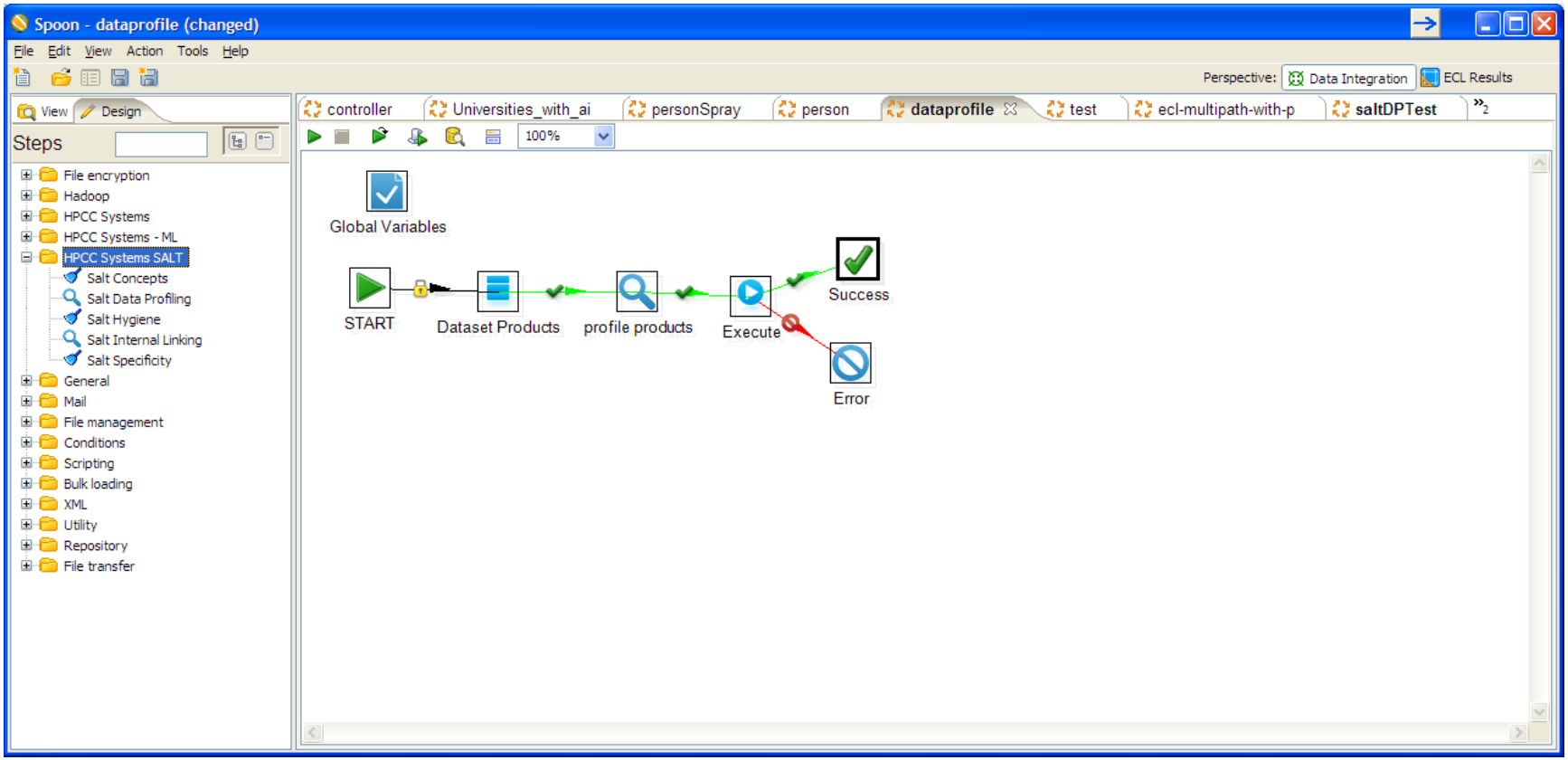
```
Join join = new Join();
join.setName("myRecSet");
join.setJoinCondition("left.theID"= "+"right.theID");
join.setJoinType("INNER");
join.setLeftRecordSet("MyLeftRecordSet");
join.setRightRecordSet("MyRightRecordSet");
String joinResults = join.ecl();
```

Resulting ECL Code

```
myRecSet := join(MyLeftRecordSet,MyRightRecordSet,left.theID = right.theID, INNER);
```

Java Integration: Java ECL API – SALT Code Generator

Data Profile Example



Java Integration: Java ECL API – SALT Code Generator

Data Profile Example Result Set

Try the demo:
http://hpccsystems.com/demos/data-profiling-demo
Get the source code:
https://github.com/hpcc-systems/java-ecl-api

Summary Report for 485 records

File Edit View Action Tools Help

OPEN FILE

Preview data returned from HPCC

SpoonDataProfile W20130812-122454 | SpoonDataProfile W20130812-122549

View Workunit in the Default Web Browser

Data Summary | Optimized Layout | Data Profile

FieldNumber	FieldName	Cardinality	Length	Words	Characters	Patterns	Frequent Terms
1	brand_name	9	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
2	cases_per_pallet	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
3	gross_weight	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
4	low_fat	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
5	net_weight	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
6	product_class_id	3	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
7	product_id	2	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
8	product_name	5	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
9	recyclable_package	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
10	shelf_depth	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
11	shelf_height	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
12	shelf_width	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
13	sku	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
14	srp	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show
15	units_per_case	1	[+]Show	[+]Show	[+]Show	[+]Show	[+]Show

Python Integration

Python Integration : Code Example

ECL Code Snippet using EMBED(Python)

```
IMPORT Python;

EXAMPLE 1:

SET OF INTEGER sortArrayOfIntegers (SET OF INTEGER val) := EMBED(Python)
   return sorted (val)    // Python code here
ENDEMBED;

arr := sortArray ([-111, 0, 113, 12, 45, 5]);
arr[3];   // Returns 5

EXAMPLE 2:

SET OF STRING sortArrayOfString (SET OF STRING val) := EMBED(Python)
   return sorted (val)    // Python code here
ENDEMBED;

result := sortArrayOfString (['red','green','yellow']);
result[1];   // Returns ‘green’
```

What's next?

- Version 5.0 Beta will be out soon.
- Ongoing R&D (5.2 and beyond):
  - Level 3 BLAS support
  - More Machine Learning related algorithms
  - Heterogeneous/hybrid computing (FPGA, memory computing, GPU)
  - Knowledge Engineering Language driving ML
  - General usability enhancements (new abstractions, enhanced user interfaces)
  - Additional capabilities for embedded languages
  - More integration with other tools and systems

Useful links

HPCC Systems
http://hpccsystems.com

- LexisNexis Open Source HPCC Systems Platform: http://hpccsystems.com
- Free Online Training: http://learn.lexisnexis.com/hpcc
- Machine Learning portal: http://hpccsystems.com/ml
- The HPCC Systems blog: http://hpccsystems.com/blog
- Community Forums: http://hpccsystems.com/bb
- Our GitHub portal: https://github.com/hpcc-systems
- JIRA: https://track.hpccsystems.com

Upcoming Event!

Join us during Big Data Atlanta Week for a Meetup!

EDA Toolkit for Data Scientists
May 6, 2014 at 4:30pm
LexisNexis Alpharetta office,
1000 Alderman Drive

RSVP:
http://www.meetup.com/Atlantas-Big-Data-Week-2014/events/169591702/

Trivia!

Answer the question and win a prize!

- What is the name of the data refinery engine that provides batch oriented data manipulation?
- What is the name of the data delivery engine that provides real-time analytics?
- What is the query processing language supporting the HPCC Systems platform?
- What can be used to generate ECL code for automating data profiling, parsing and cleansing?
- Name three machine learning algorithms supported in ECL-ML.
- What does KEL stand for?

Questions???

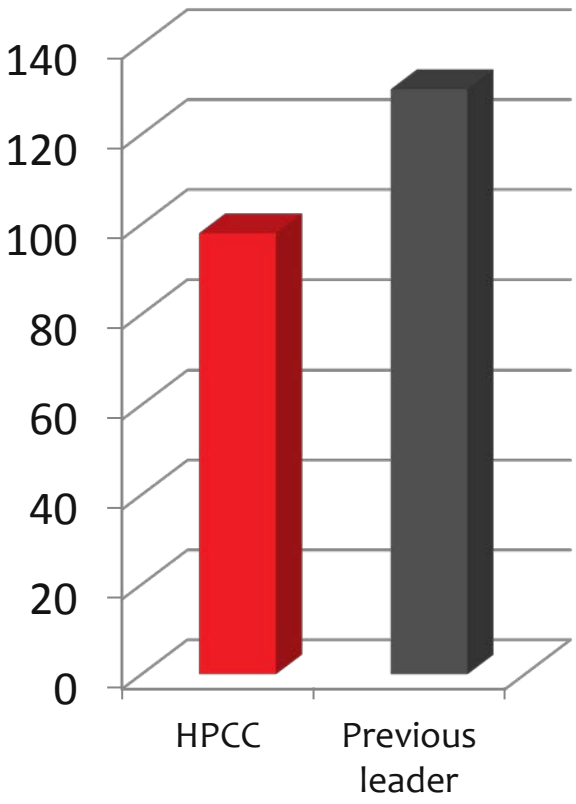
HPCC Systems
http://hpccsystems.com

Appendix

Terasort Benchmark results

HPCC Systems
http://hpccsystems.com

Execution Time (seconds)



Productivity

3 Lines of ECL

```
// Perform global terasort
rec := record
      string10  key;
      string10  seq;
      string80  fill;
       end;
in :=DATASET('~nhtest::terasort1',rec,FLAT);
OUTPUT(SORT(in,key,UNSTABLE),,'nhtest::terasort1out',overwrite);
//End
```

700+ Lines of Java MapReduce Code

```
  }
  abstract int findPartition(Text key);
  abstract void print(PrintStream strm) throws IOException;
  int getLevel() {
    return level;
  }
}

/**
 * An inner trie node that contains 256 children based on the next
 * character.
 */
static class InnerTrieNode extends TrieNode {
  private TrieNode[] child = new TrieNode[256];

  InnerTrieNode(int level) {
    super(level);
  }
  int findPartition(Text key) {
    int level = getLevel();
    if (key.getLength() <= level) {
      return child[0].findPartition(key);
    }
    return child[key.getBytes()[level] & 0xff].findPartition(key);
```

Space/Cost

Medicaid Case Study

HPCC Systems
http://hpccsystems.com

Scenario

Proof of concept for Office of the Medicaid Inspector Generation (OMIG) of large Northeastern state. Social groups game the Medicaid system which results in fraud and improper payments.

Task

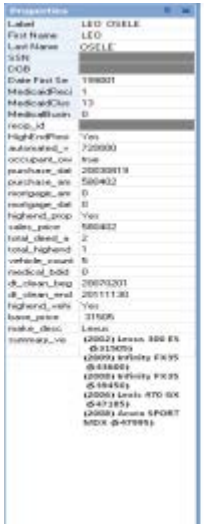
Given a large list of names and addresses, identify social clusters of Medicaid recipients living in expensive houses, driving expensive houses.

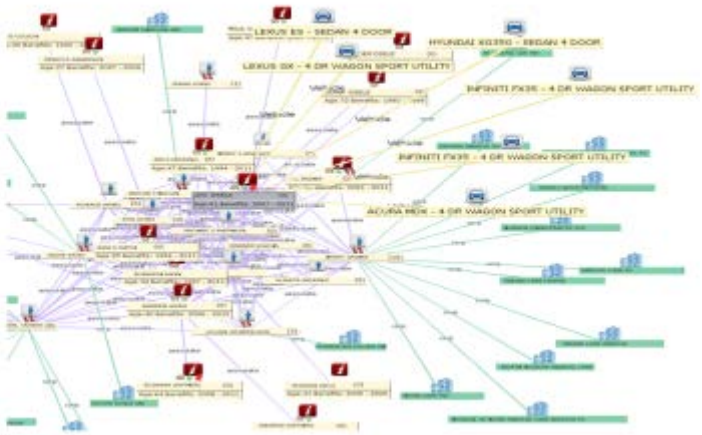
Result

Interesting recipients were identified using asset variables, revealing hundreds of high-end automobiles and properties.

Leveraging the Public Data Social Graph, large social groups of interesting recipients were identified along with links to provider networks.

The analysis identified key individuals not in the data supplied along with connections to suspicious volumes of "property flipping" potentially indicative of mortgage fraud and money laundering

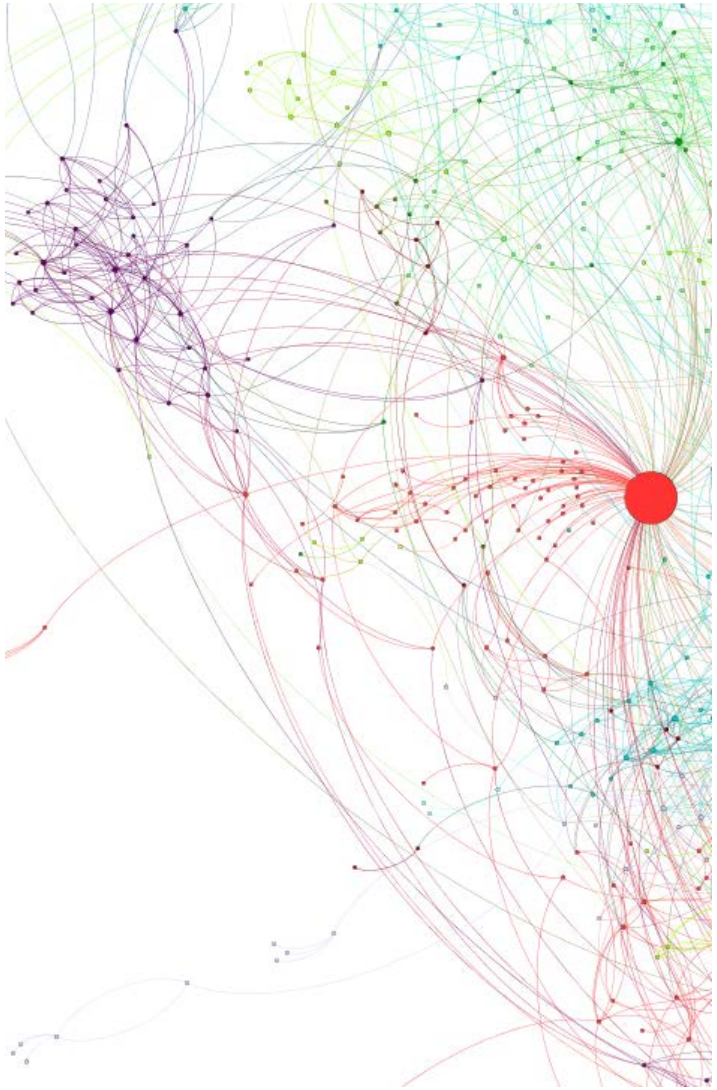




Make Description	#	Make Description	#
Mercedes-Benz	46	Chevrolet	2
Lexus	41	Hummer	2
BMW	27	Jeep	2
Infiniti	13	Nissan	2
Acura	9	Toyota	2
Lincoln	8	Aston Martin	1
Audi	7	Bentley	1
Land Rover	7	Cadillac	1
Porsche	6	GMC	1
Jaguar	5	Honda	1
Mercedes Benz	3	Volkswagen	1
Saab	3	Volvo	1

Social Graph Analytics - Collusion

HPCC Systems
http://hpccsystems.com



- LexisNexis Public Data Social Graph (PDSG)
  - Public Data relationships.
  - High Value relationships for Mapping trusted networks.
- Large Scale Data Fabrication and Analytics.
  - Thousands of data sources to ingest, clean, aggregate and link.
  - 300 million people, 4 billion relationships, 700 million deeds.
  - 140 billion intermediate data points when running analysis.
- HPCC Systems from LexisNexis Risk Solutions
  - Open Source Data Intensive high performance supercomputer. (http://hpccsystems.com)
- Innovative Examples leveraging the LexisNexis PDSG
  - Healthcare.
    - Medicaid\Medicare Fraud.
    - Drug Seeking Behavior
  - Financial Services.
    - Mortgage Fraud.
    - Anti Money Laundering.
    - “Bust out” Fraud.
- Potential Collusion (The value in detecting non arms length transactions)

Connected Legal Content Universe

HPCC Systems
http://hpccsystems.com