Distributed Machine Learning

*Powered by HPCC Systems®*

Flavio Villanustre
VP Technology, LexisNexis Risk Solutions

September 21, 2016
HPCC Systems Technology: Big Data Is Our Core Competency

**SPEED**
- Scales to extreme workloads quickly and easily
- Increases speed of development leads to faster production/delivery
- Improves developer productivity

**CAPACITY**
- Enables massive joins, merges, transformations, sorts, or tough N^2 problems
- Increases business responsiveness
- Accelerates creation of new services via rapid prototyping capabilities
- Offers a platform for collaboration and innovation leading to better results

**COMPLEX PROCESSING**
- Disambiguates entities with a high level of speed and accuracy
- Constructs graphs from complex, large data sets for easier data analytics
- Enables graph traversal to recognize areas of hidden value
- Identifies important attributes that contribute to predictive models

**COST SAVINGS**
- Leverages commodity hardware so fewer people can do much more in less time
- Uses IT resources efficiently via sharing and higher system utilization
The Data Centric Approach

A single source of data is insufficient to overcome inaccuracies in the data.

The holes are inaccuracies found in the data.

Our platform is built on the premise of absorbing data from multiple data sources and transforming them to a highly intelligent social network graphs that can be manipulated to extract the non-obvious value.

The holes in the core data have been eliminated.
Our Solutions Are Powered by HPCC Systems at Their Core

**Big Data**

- **Public Records**
- **Proprietary Data**
- **News Articles**
- **Unstructured Records**
- **Structured Records**

**Unstructured and Structured Content**
- Over 4 petabytes of content
- 50 billion records
- 10,000 sources
- 7.5 billion unique name and address combinations

**High Performance Computing Cluster Platform (HPCC)**
- Grid computing
- Data-centric language (ECL)
- Integrated delivery system that offers data plus analytics

**Analysis Applications**
- Financial Services
- Government
- Health Care
- Insurance
- Legal
- Retail
- Scientific Technical Medical
- Exhibitions

**Open Source Components**

**Key Capabilities**
- Data and analytics
- Identity verification and authentication
- Fraud detection and prevention
- Investigation
- Screening
- Receivables management

**Entity Resolution**
**Link Analysis**
**Clustering Analysis**
**Complex Analysis**
Data Flow Oriented Big Data Platform

Thor (Data Lake)
- Shared Nothing MPP Architecture
- Commodity Hardware
- Batch ETL and Analytics

ROXIE (Query)
- Shared Nothing MPP Architecture
- Commodity Hardware
- Real-time Indexed Based Query
- Low Latency, Highly Concurrent and Highly Redundant

Raw data from several sources

Batch requests for scoring and analytics

ECL
- Easy to use
- Implicitly Parallel
- Compiles to C++
The Technology Stack

**STRIKE**

- Large scale data integration
- Probabilistic linking
- Entity disambiguation and resolution

**SALT**

- Batch oriented Big Data processing and analytical engine
- Machine learning supervised model training
- Clustering

**Thor**

- Horizontally scalable and fault tolerant real-time disk-based retrieval and analytics
- Horizontally scalable in-memory analytics

**ROXIE**

- Seamless data integration with hundreds of data stores
- Real-time data ingest
- Flexible stream processing

**Interlok**

- Graph/Network data models
- Complex queries based on n-degree relationships and attributes
- Highly efficient

**KEL**

- Dataflow oriented declarative data programming language
- Compiles to C++ for optimal performance
- High expressivity and conciseness

**ECL**
STRIKE Technology Component View

Integration System

- Data Integration
  - Connect
  - Integrate
  - Schedule
  - Transform

Standardization & Aggregation System

- Standardization
  - Clean
  - Profile
  - Normalize

Aggregation

- Master Data Creation
- Relationship Analysis
- Predictive Analysis
- Business Intelligence

Visualization System
Master Data Management with SALT

From disparate data, to clustering, to showing relationships
The acronym stands for “Scalable Automated Linking Technology”

Entity disambiguation using Inference Techniques

Templates based ECL code generator

Provides for automated data profiling, parsing, cleansing, normalization and standardization

Sophisticated specificity and relatives based linking and clustering

Data Sources

Profiling → Parsing → Cleansing → Normalization → Standardization

1. Data Preparation Processes (ETL)

Matching Weights & Threshold Computation → Blocking/Searching → Weight Assignment & Record Comparison → Record Match Decision

Linked Data File

2. Record Linkage Process
SALT’s Superior Linking Technology

SALT eliminates **FALSE NEGATIVES** using probabilistic learning

1. Flavio Villanustre, Atlanta
2. Javio Villanustre, Atlanta

**SALT**

MATCH — the system has learnt that “Villanustre” is specific because the frequency of occurrence is small and there is only one present in Atlanta

ERROR

NO MATCH — because the rules determine that “Flavio” and “Javio” are not the same

SALT eliminates **FALSE POSITIVES** using probabilistic learning

1. John Smith, Atlanta
2. John Smith, Atlanta

**SALT**

MATCH — 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

NO MATCH — because the rules determine that “John Smith” and the city for both the records match
KEL — an abstraction for network/graph processing

- Declarative model: describe what things are, rather than how to execute
- High level: vertices and edges are first class citizens
- A single model to describe graphs and queries
- Leverages Thor for heavy lifting and ROXIE for real-time analytics
- Compiles into ECL (and ECL compiles into C++, which compiles into assembler)
# A Comparison: KEL and ECL

<table>
<thead>
<tr>
<th>ECL</th>
<th>KEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generates C++ (1-&gt;100)</td>
<td>Generates ECL (1-&gt;12)</td>
</tr>
<tr>
<td>Files and Records</td>
<td>Entities and associations</td>
</tr>
<tr>
<td>Detailed control of data format</td>
<td>Loose control of input format; none of processing</td>
</tr>
<tr>
<td>Can write graph and statistical algorithms</td>
<td>Major algorithms built in</td>
</tr>
<tr>
<td>Thor/Roxie split by human design</td>
<td>Thor/Roxie split by system design</td>
</tr>
<tr>
<td>Solid, reliable and mature</td>
<td>R&amp;D</td>
</tr>
</tbody>
</table>
Machine Learning with HPCC Systems
Machine Learning on HPCC Systems

- 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: https://hpccsystems.com/ml
ECL-ML: extensible ML

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 (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 Systems

• 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!
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

```plaintext
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
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.

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'));
```

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

Search Term: sentiment analysis
Type: Absolute Chart

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'));
```
Resources

- Portal: http://hpccsystems.com
- SALT: https://hpccsystems.com/enterprise-services/purchase-required-modules/SALT
- KEL: https://hpccsystems.com/download/free-modules/kel-lite
- Machine Learning: http://hpccsystems.com/ml
- Online Training: http://learn.lexisnexis.com/hpcc
- HPCC Systems Blog: http://hpccsystems.com/blog
- Our GitHub portal: https://github.com/hpcc-systems
- Community Forums: http://hpccsystems.com/bb
- Case Studies: https://hpccsystems.com/resources/case-studies
Appendix: Customer Use Cases
Example #1: Graph Analysis
Bust-Out Fraud

SCENARIO
• Perpetrators typically apply for credit 4 to 24 months before busting out
• Hard to predict at an individual level

Apply for Credit → Build good credit history → Obtain additional credit → Draw down all available credit → Disappear

More than $2 billion in annual losses
Example #1: Graph Analysis

The Solution Highlights

SCENARIO

• Financial company hypothesized that organized groups were targeting them and desired to tackle bust-out fraud at a social level

• Although some risk was not individually large, the company wanted to ascertain where that risk was growing without them realizing the connections
Example #1: Graph Analysis
The Solution Highlights

DATA & ANALYSIS

• 5 million accounts flagged with active, known fraud, charge offs and preemptively closed tags
• We investigated every address, asset and business connection attached to the accounts

OUR APPROACH

• Standardized input, applied LexID and joined to our Social Graph (5 million into 4 billion relationships)
• Calculated grouped social aggregates, scored and rank ordered the result
Example #1: Graph Analysis

Results Overview

1. Known Fraud
   - # in 1 degree | # accounts
     0 | 4,055,099
     1 | 39,266
     2 | 638
     3 | 31
     4 | 2

   - 31 associated with 3 known fraud accounts

2. Charge Off
   - # in 2 degrees | # accounts
     0 | 4,079,930
     1 | 14,894
     2 | 212

   - 212 associated with 2 charge accounts

3. Preemptively Closed
   - # in 1 degree | # accounts
     0 | 3,834,840
     1 | 239,472
     2 | 19,026
     3 | 1,322
     4 | 194
     5 | 71
     6 | 21
     7 | 10
     8 | 7
     9 | 14
    10 | 9
    11 | 7
    12 | 7
    13 | 14
    14 | 14
    15 | 3
    16 | 2
    17 | 1
    21 | 2

   - 3 accounts associated with 15 preemptively closed accounts
Example #1: Graph Analysis
Account Overview

Aggregated group behavior variables for a single account

- Flatten the graph
- Calculate aggregate group behavior measurements
- Drive predictive analytics at a granular level using graph variables

**LexID:** 312873  
**Jack Johnson**  
**Cluster ID:** 1214379

<table>
<thead>
<tr>
<th># Network Neighborhood</th>
<th>81</th>
</tr>
</thead>
<tbody>
<tr>
<td># 1&lt;sup&gt;st&lt;/sup&gt; Degrees</td>
<td>8</td>
</tr>
<tr>
<td># 2&lt;sup&gt;nd&lt;/sup&gt; Degrees</td>
<td>72</td>
</tr>
<tr>
<td>Cohesivity</td>
<td>1.67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Active Accounts</th>
<th>Total</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree 0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Degree 1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Degree 2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Fraud Accounts</th>
<th>&lt;=2 Degrees</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Degree 1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Degree 2</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Charge Off Accounts</th>
<th>&lt;=2 Degrees</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Degree 1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Degree 2</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># Preemptive Close Accounts</th>
<th>&lt;=2 Degrees</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Degree 0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Degree 1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Degree 2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**SCORE:** 6.6

- Small network
- Fairly tightly connected
- 3 active accounts within the network that are known fraud
- 9 accounts within that network that are known fraud
- 1 account within the network that’s been preemptively closed
Example #2: Graph Analysis
Insurance Collusion

**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.
Example #2: Graph Analysis
Insurance Collusion

TASK
• After adding the Link ID to the carrier data, our HPCC technology then added 2 additional degrees of relatives

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
Smart Hard Hat Ecosystem

THE CHALLENGE

- 4,000 workers die and millions injured annually while working on the industrial floor
- Very high cost for maintaining safety for businesses
Smart Hard Hat Ecosystem

THE SOLUTION

• Equip workers’ hats with smart sensor technology
• Central real-time processing of (high volume) information with real-time alerting capability (HPCC Systems)
• Customizable dashboards, rules framework and data workflow frameworks (HPCC Systems)
• Predictive modeling and analytics (HPCC Systems)

THE OUTCOME: Produced an industrial wearable that uses IoT and wireless communications systems to protect and empower industrial workers.
Driver Behavior with Smart Telematics

THE CHALLENGE

- High cost of insurance
- High car accident rates
- Lack of tools to analyze driver behavior
THE SOLUTION

- Telematics smart phone application
- Central system to collect (very large) data and perform analytics (HPCC Systems)
- Journey based feedback to all drivers to advice and correct behavior (HPCC Systems)
- Insurance enrollment to reduce premiums

THE OUTCOME:

- Recommend corrections to driver behavior that would avoid accidents
- Reduce overall Insurance costs
- Correlate information from drivers data traversing the same path to create an understanding of predictable actions
- Examples include periods of traffic congestion, problem areas in the path and hazard detection
Lack of deep insight into e-learning activities (training patterns, suspicious activities, problem tests and developing career paths)

Collection and analysis of large quantities of data (500,000 users and 3 million courses)
THE OUTCOME: Prevent fraudulent use of the e-learning systems and creation of a recommender system for course ware

THE SOLUTION

• Used Big Data technologies to collect, store and analyze the data (HPCC Systems)
Contextual Marketing

THE CHALLENGE

Understanding an individual customer’s behavior based on past actions

Technical problem

- Huge volumes of data based on observed cell phone Wi-Fi
- Apply advanced machine learning techniques
Contextual Marketing

THE SOLUTION

- Central analytics system to collect and analyze data (HPCC Systems)
- Leverage parallel algorithms to perform analytics on large quantities of data (HPCC Systems)

THE OUTCOME:

Created a platform to process any location specific telecom data that can be analyzed rapidly to gauge consumer behavior and in turn help drive context-based marketing
Predict Passenger Volumes in Airports

THE CHALLENGE

- How to interpret 100’s millions of location points while adjusting to flight schedule changes
- Complex clustering algorithm requirements
- Understand passenger behaviors and interaction of local areas of activity
HPCC used to solve Big Data challenges

- Raw data to refined data
- Clustering analysis
- Forecasting

THE OUTCOME:
Better passenger experience and better airport planning
SciVal

THE CHALLENGE

- Fast insight into evidence based research data
- Scopus data covering 21,000 titles from 5,000 publishers that's updated weekly
- Data from 4,600 research institutions and 220 countries
- Real-time user specific data slicing
THE SOLUTION

- Clean, standardize and build Analytics Queries (HPCC Thor) from 40+ TB of data on a weekly basis
- Support rapid query interface to support both ad hoc queries and canned queries (HPCC ROXIE)

THE OUTCOME: 🎉

Two enormous benefits:

- Users can customize their own visualizations, which are generated in just a few seconds
- HPCC Systems works in 2 modes:
  1. offline crunching of huge, pre-defined requests
  2. smaller calculations in real-time on customer generated data slices