• DEMO TALK •

The Kit and Kaboodle for Big Data and Data Science

Bob Foreman
Sr Software Engineer
LexisNexis
Risk Solutions

Hugo Watanuki
Sr Technical Support Engineer
LexisNexis
Risk Solutions

ODSC EAST
VIRTUAL TRAINING CONFERENCE

March 30th – April 1st

HPCC Systems:
End to End Data Lake Management

Completely free

open source data lake solution

Out of the box capabilities for consistency and

ease of use

Less coding

and more using (even though we love to code)

We are your one stop shop for all your data integration, querying and analytical needs

10 YEARS

HPCC SYSTEMS®

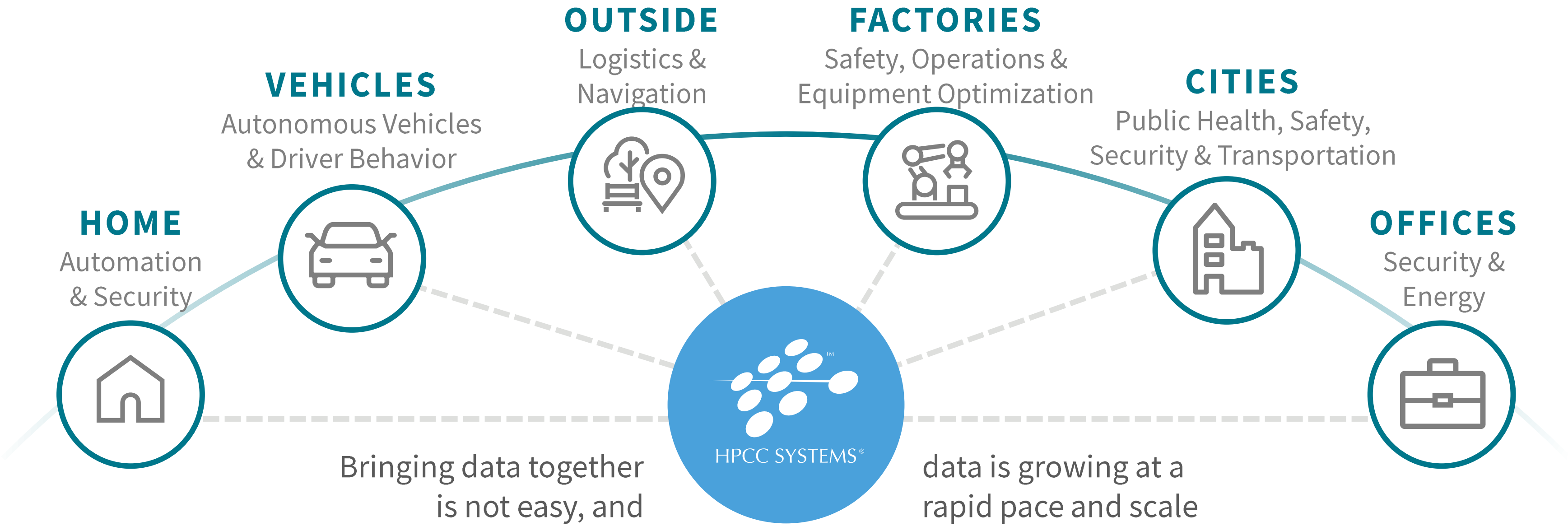
OPEN SOURCE

A Brief History of HPCC!

Why does HPCC Systems exist?

- ✓ It was NOT developed with the idea of selling the technology to anybody else!
- ✓ It was all created only to solve some of the data-handling problems that we encountered as we were developing our products.
- ✓ HPCC defined is a *distributed data parallel processing* platform.

A platform purpose-built for high-speed data engineering



OUTSIDE
Logistics & Navigation
FACTORIES
Safety, Operations & Equipment Optimization
VEHICLES
Autonomous Vehicles & Driver Behavior
CITIES
Public Health, Safety, Security & Transportation
HOME
Automation & Security
OFFICES
Security & Energy
HPCC SYSTEMS®

Bringing data together is not easy, and data is growing at a rapid pace and scale

A processing platform is vital for bringing all your data together across all verticals

HPCC Systems Evolution

2001

Original version of HPCC Systems released

2011

Open source Apache license and code release to GitHub

Exceeded market-leading performance benchmark achieved

2012 – 16

Continuous **QUALITY-FOCUSED** improvements

Better support and training with improved integration — faster and easier to use

2017-2020

Improved processing architecture

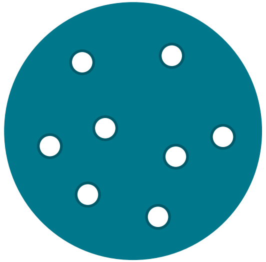
IoT enabled

More Bundles and ML Expansion!

HPCC SYSTEMS®

The Data Centric Approach

A single source of data is insufficient to overcome inaccuracies

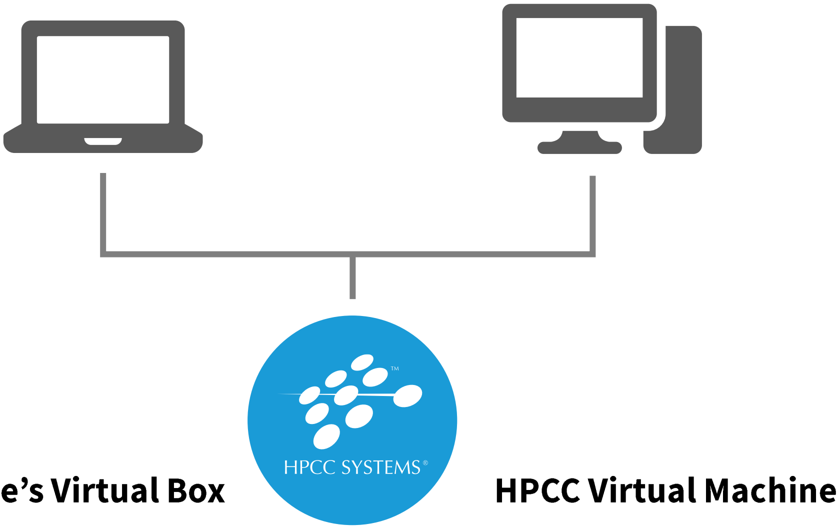


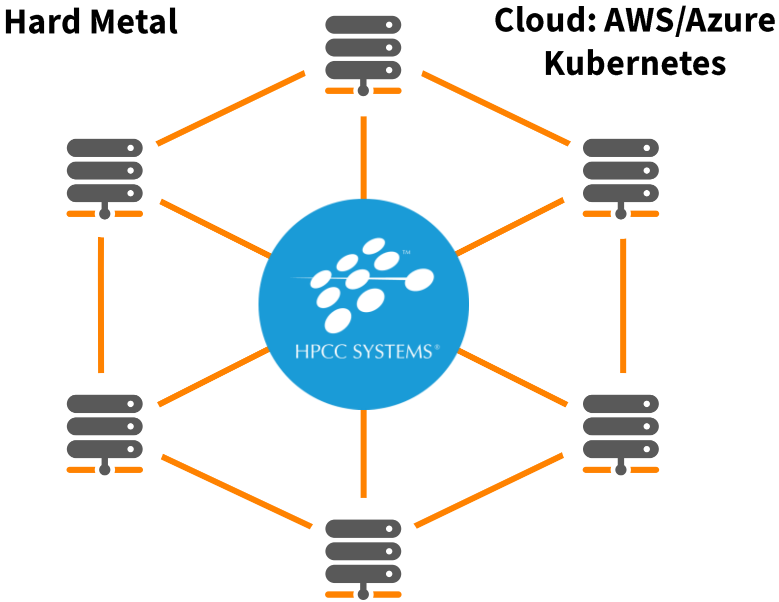
Our platform is built on the premise of absorbing data from **many data sources** and transforming them to **actionable smart data**

Scale from Small to Big

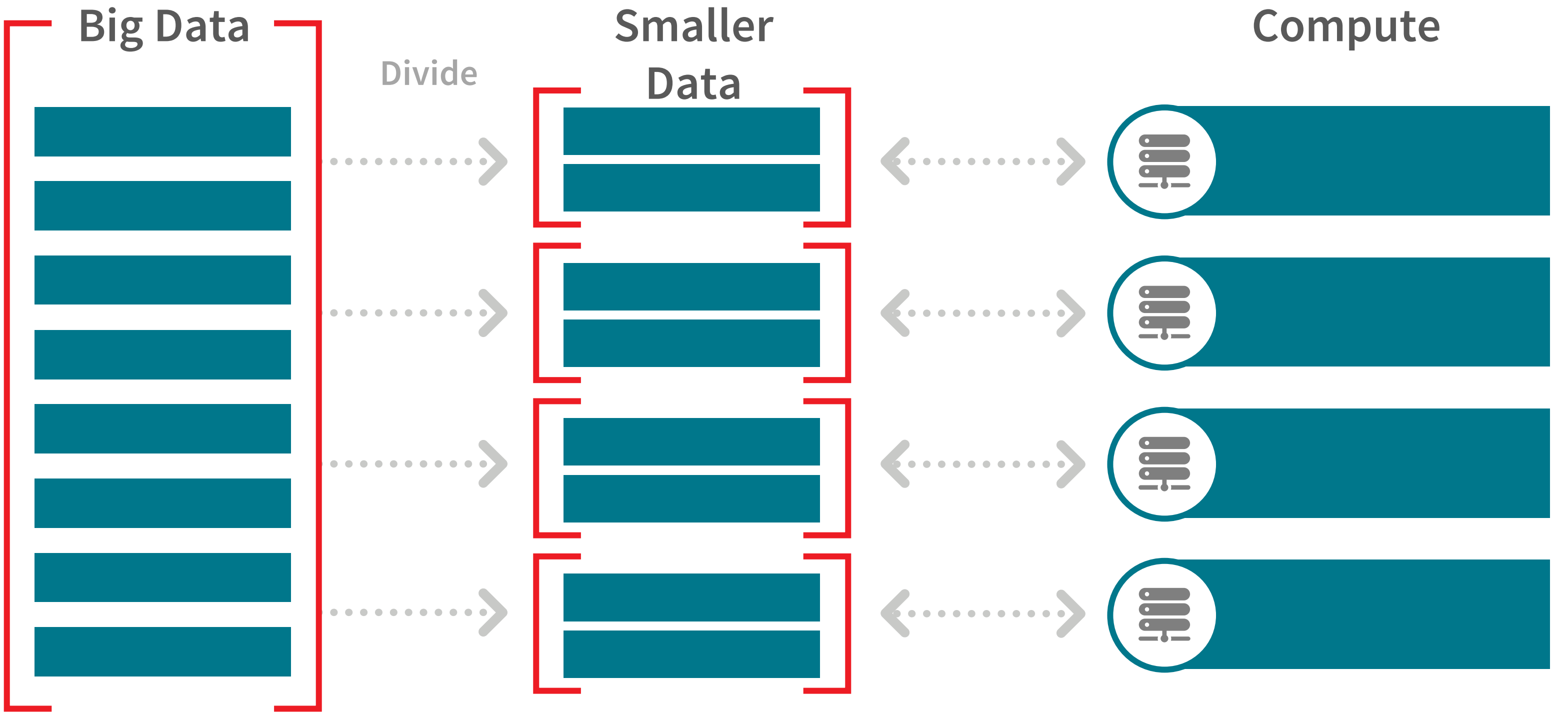
The stack can run on a single laptop or desktop.

In more sophisticated cases, HPCC Systems run *clusters*, hundreds of servers working as a single processing entity, to transform and deliver big data.

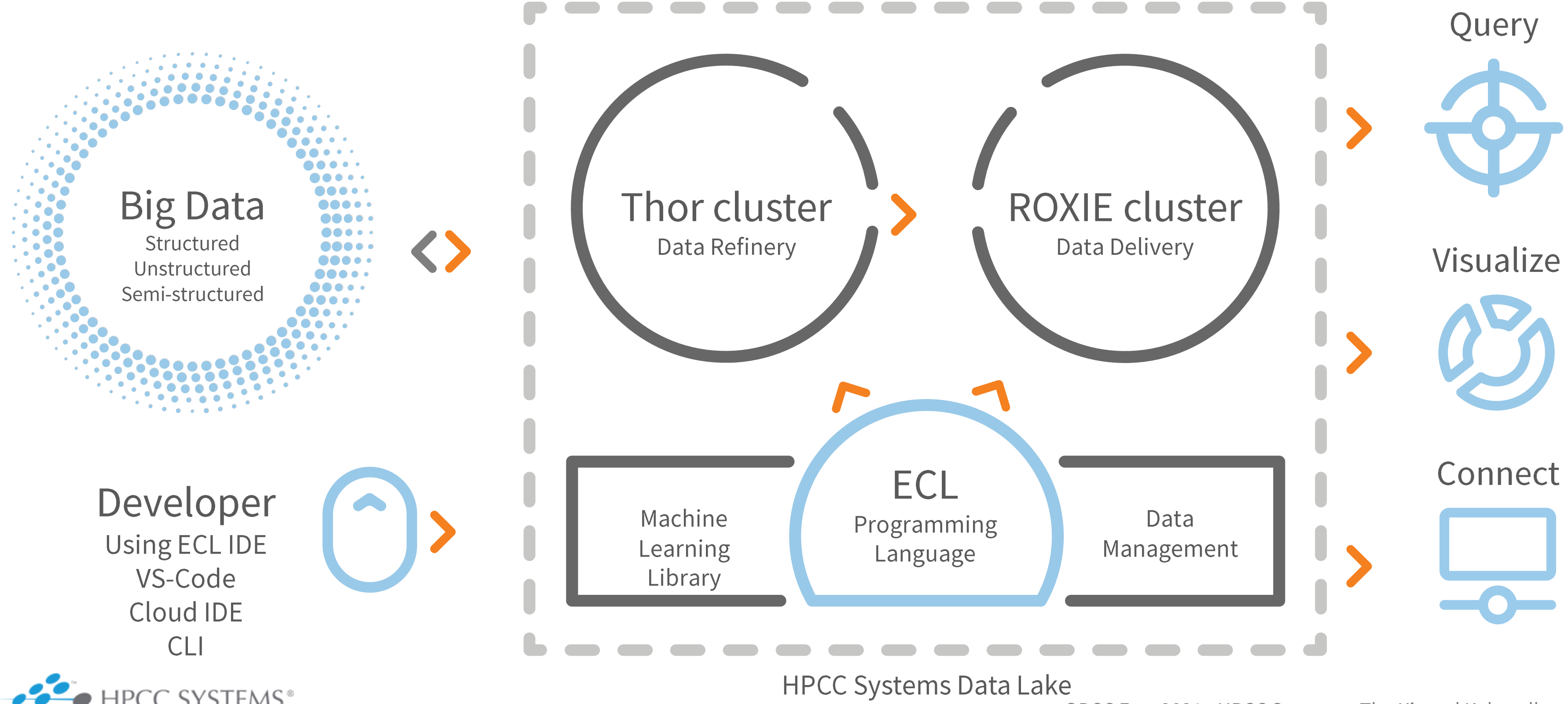




Anatomy of a Big Data Processing System



The HPCC Systems Components



HPCC SYSTEMS®

Technology — The Open Source Stack

Thor: Data Refinery Cluster

Extraction, loading, cleansing, transforming, linking and indexing

ROXIE: Data Delivery Engine

Rapid data delivery cluster with high-performance online query delivery for big data

Data Management Tools

Data profiling, cleansing, snapshot data updates, consolidation, job scheduling and automation

Machine Learning Library

Linear regression, logistic regression, decision trees and random forests

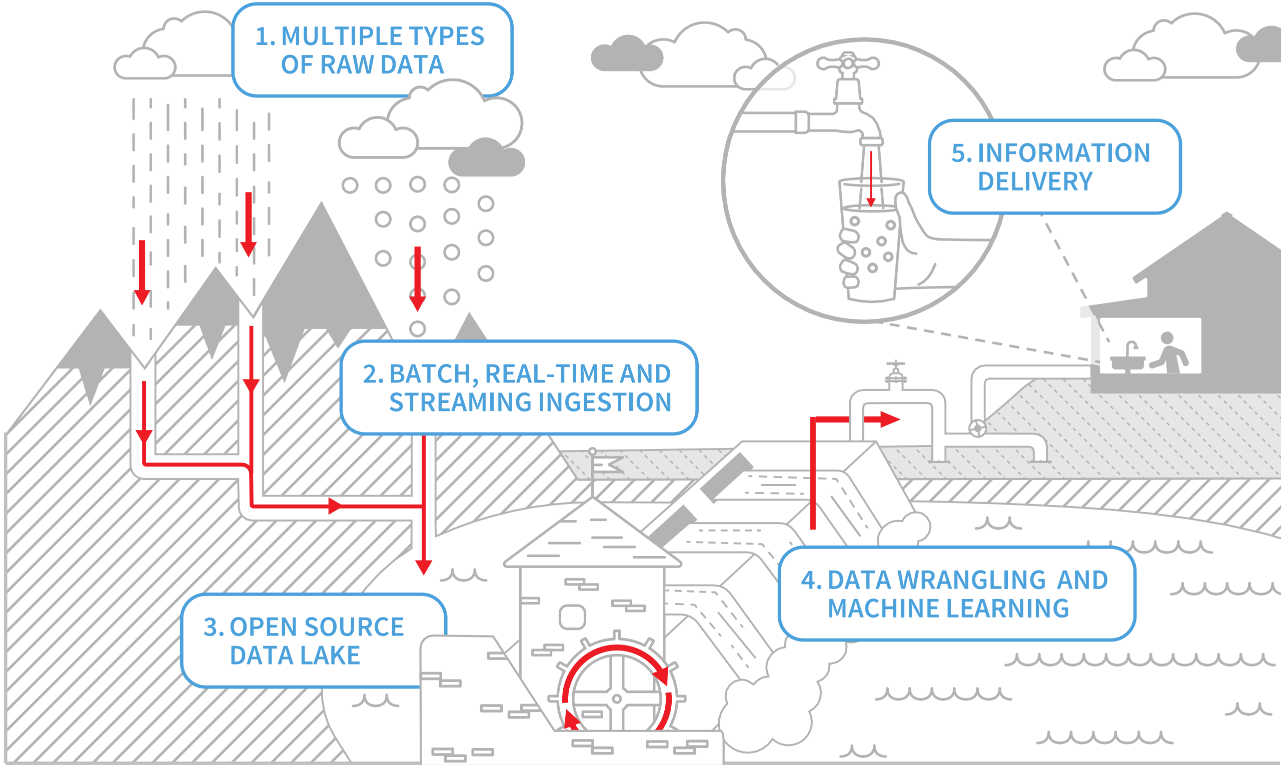
Connectivity & Third-Party Tools

New plugins to help integrate third party tools with the HPCC Systems platform

Key aspects of our data lake solution

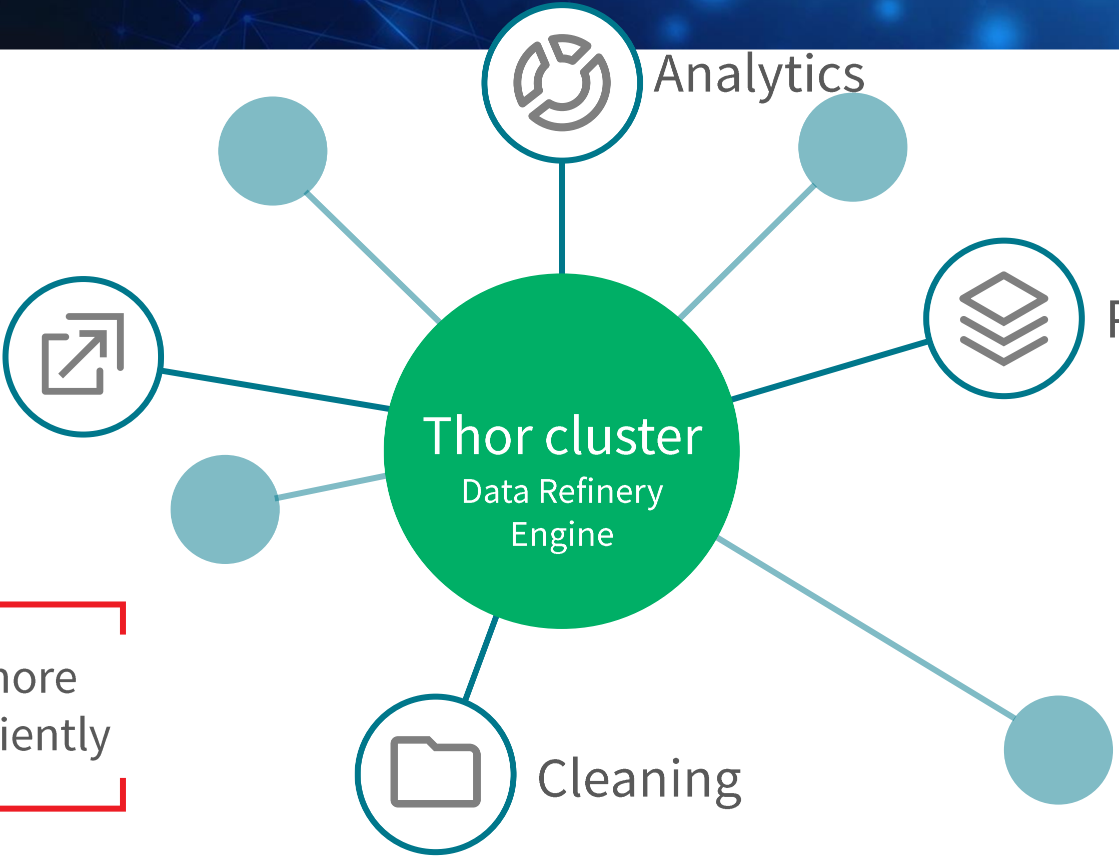
The HPCC Systems advantage

- Open source data lake platform
- Batch, real-time and streaming data ingestion
- Built-in data enhancement and Machine Learning APIs
- Scalable to many petabytes of data
- Runs on commodity hardware and in the cloud
- Increased responsiveness to customers and stakeholders


1. MULTIPLE TYPES OF RAW DATA
2. BATCH, REAL-TIME AND STREAMING INGESTION
3. OPEN SOURCE DATA LAKE
4. DATA WRANGLING AND MACHINE LEARNING
5. INFORMATION DELIVERY

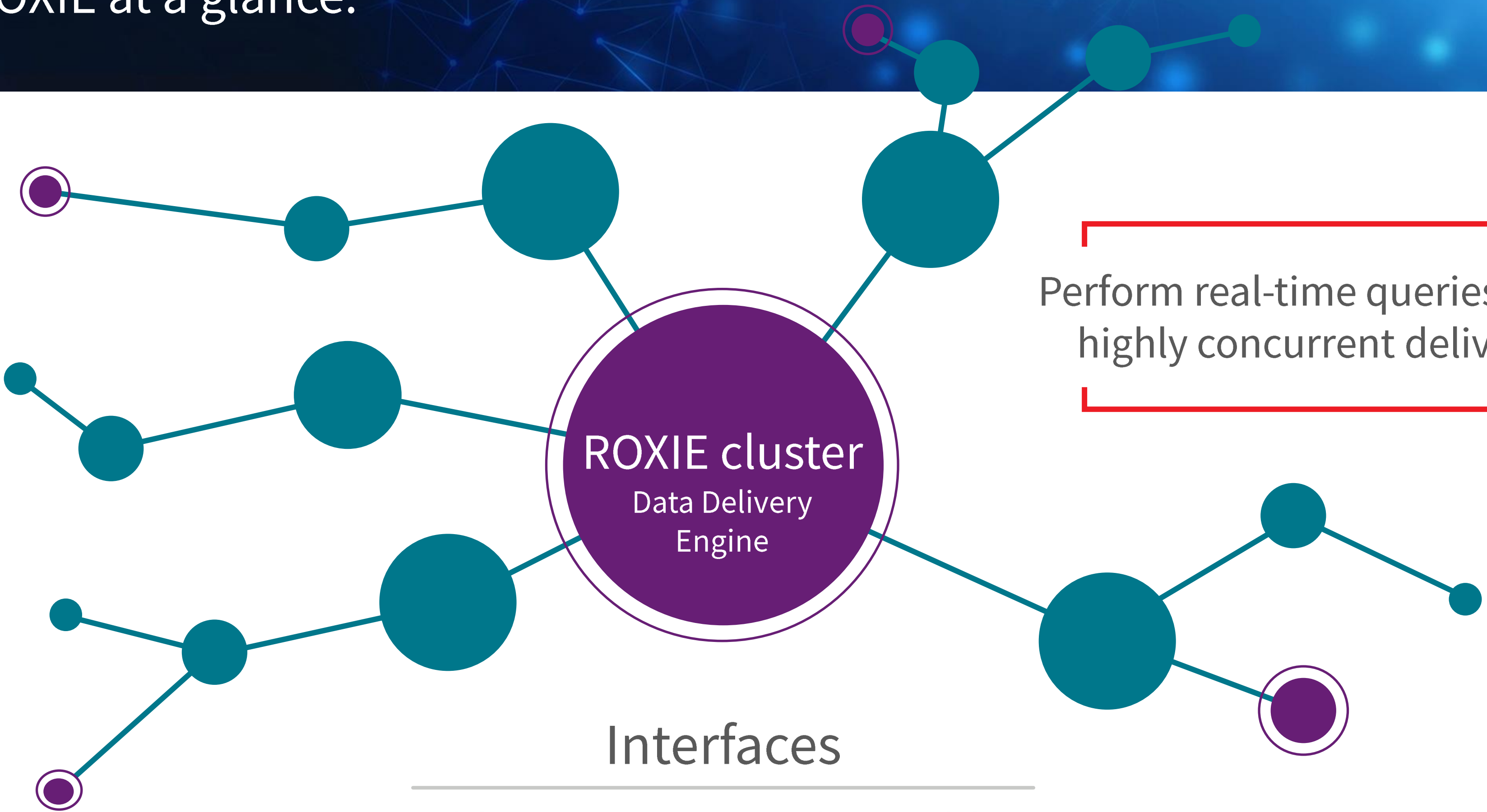
HPCC SYSTEMS®

THOR at a glance:



Manage data more quickly and efficiently

ROXIE at a glance:



Perform real-time queries with highly concurrent delivery

Interfaces

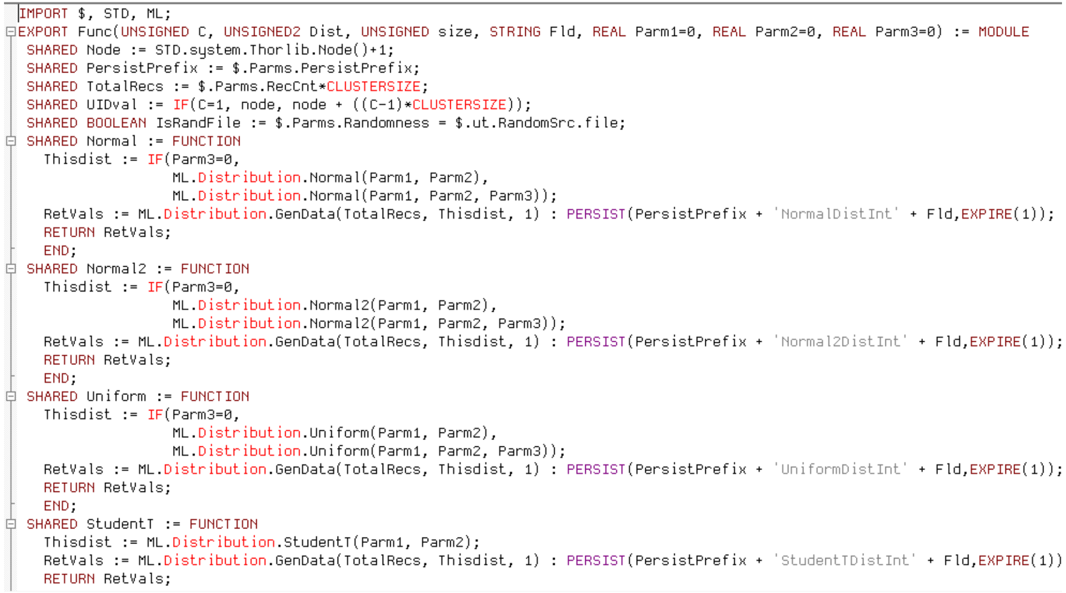
JSON XML REST SQL

An Introduction to ECL

ECL

Enterprise Control Language

- Transparent and implicitly parallel programming language
- Both powerful and flexible



```
IMPORT $, STD, ML;
EXPORT Func(UNSIGNED C, UNSIGNED2 Dist, UNSIGNED size, STRING Fld, REAL Parm1=0, REAL Parm2=0, REAL Parm3=0) := MODULE
 SHARED Node := STD.system.Thorlib.Node()+1;
 SHARED PersistPrefix := $.Parms.PersistPrefix;
 SHARED TotalRecs := $.Parms.RecCnt*CLUSTERSIZE;
 SHARED UIDval := IF(C=1, node, node + ((C-1)*CLUSTERSIZE));
 SHARED BOOLEAN IsRandFile := $.Parms.Randomness = $.ut.RandomSrc.file;
 SHARED Normal := FUNCTION
   Thisdist := IF(Parm3=0,
                  ML.Distribution.Normal(Parm1, Parm2),
                  ML.Distribution.Normal(Parm1, Parm2, Parm3));
   RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'NormalDistInt' + Fld,EXPIRE(1));
   RETURN RetVals;
   END;
 SHARED Normal2 := FUNCTION
   Thisdist := IF(Parm3=0,
                  ML.Distribution.Normal2(Parm1, Parm2),
                  ML.Distribution.Normal2(Parm1, Parm2, Parm3));
   RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'Normal2DistInt' + Fld,EXPIRE(1));
   RETURN RetVals;
   END;
 SHARED Uniform := FUNCTION
   Thisdist := IF(Parm3=0,
                  ML.Distribution.Uniform(Parm1, Parm2),
                  ML.Distribution.Uniform(Parm1, Parm2, Parm3));
   RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'UniformDistInt' + Fld,EXPIRE(1));
   RETURN RetVals;
   END;
 SHARED StudentT := FUNCTION
   Thisdist := ML.Distribution.StudentT(Parm1, Parm2);
   RetVals := ML.Distribution.GenData(TotalRecs, Thisdist, 1) : PERSIST(PersistPrefix + 'StudentTDistInt' + Fld,EXPIRE(1))
   RETURN RetVals;
```

How to do it ✗ vs. ✓ What to do

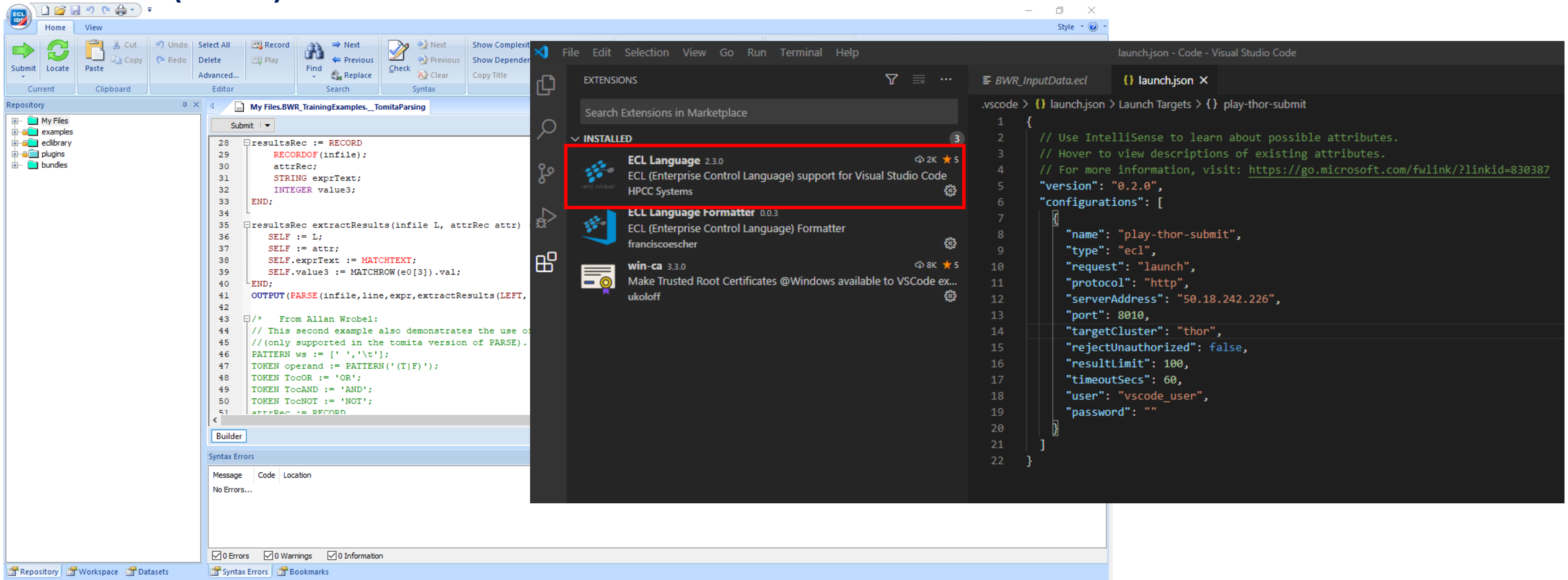
- Optimized for data-intensive operations, declarative, non-procedural and dataflow oriented
- Uses intuitive syntax which is modular, reusable, extensible and highly productive

HPCC SYSTEMS®

Integrated Development Environments

ECL IDE (Win)

Visual Studio Code (Ux/MacOS)



```
resultsRec := RECORD
     RECORDOF(infile);
     attrRec;
     STRING exprText;
     INTEGER value3;
END;

resultsRec extractResults(infile L, attrRec attr)
    SELF := L;
    SELF := attr;
    SELF.exprText := MATCHTEXT;
    SELF.value3 := MATCHROW(e0[3]).val;
END;
OUTPUT(PARSE(infile,line,expr,extractResults(LEFT,

/*   From Allan Wrobel:
// This second example also demonstrates the use o
//(only supported in the tomita version of PARSE).
PATTERN ws := [' ','\t'];
TOKEN operand := PATTERN('(T|F)');
TOKEN TocOR := 'OR';
TOKEN TocAND := 'AND';
TOKEN TocNOT := 'NOT';
attrRec := RECORD
```

ECL Language 2.3.0 — 2K ★ 5
ECL (Enterprise Control Language) support for Visual Studio Code
HPCC Systems

ECL Language Formatter 0.0.3
ECL (Enterprise Control Language) Formatter
franciscoescher

win-ca 3.3.0 — 8K ★ 5
Make Trusted Root Certificates @Windows available to VSCode ex...
ukoloff

```
{
  // Use IntelliSense to learn about possible attributes.
  // Hover to view descriptions of existing attributes.
  // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
  "version": "0.2.0",
  "configurations": [
    {
      "name": "play-thor-submit",
      "type": "ecl",
      "request": "launch",
      "protocol": "http",
      "serverAddress": "50.18.242.226",
      "port": 8010,
      "targetCluster": "thor",
      "rejectUnauthorized": false,
      "resultLimit": 100,
      "timeoutSecs": 60,
      "user": "vscode_user",
      "password": ""
    }
  ]
}
```

And CLI too! ECL.EXE

ECL IDE Features:

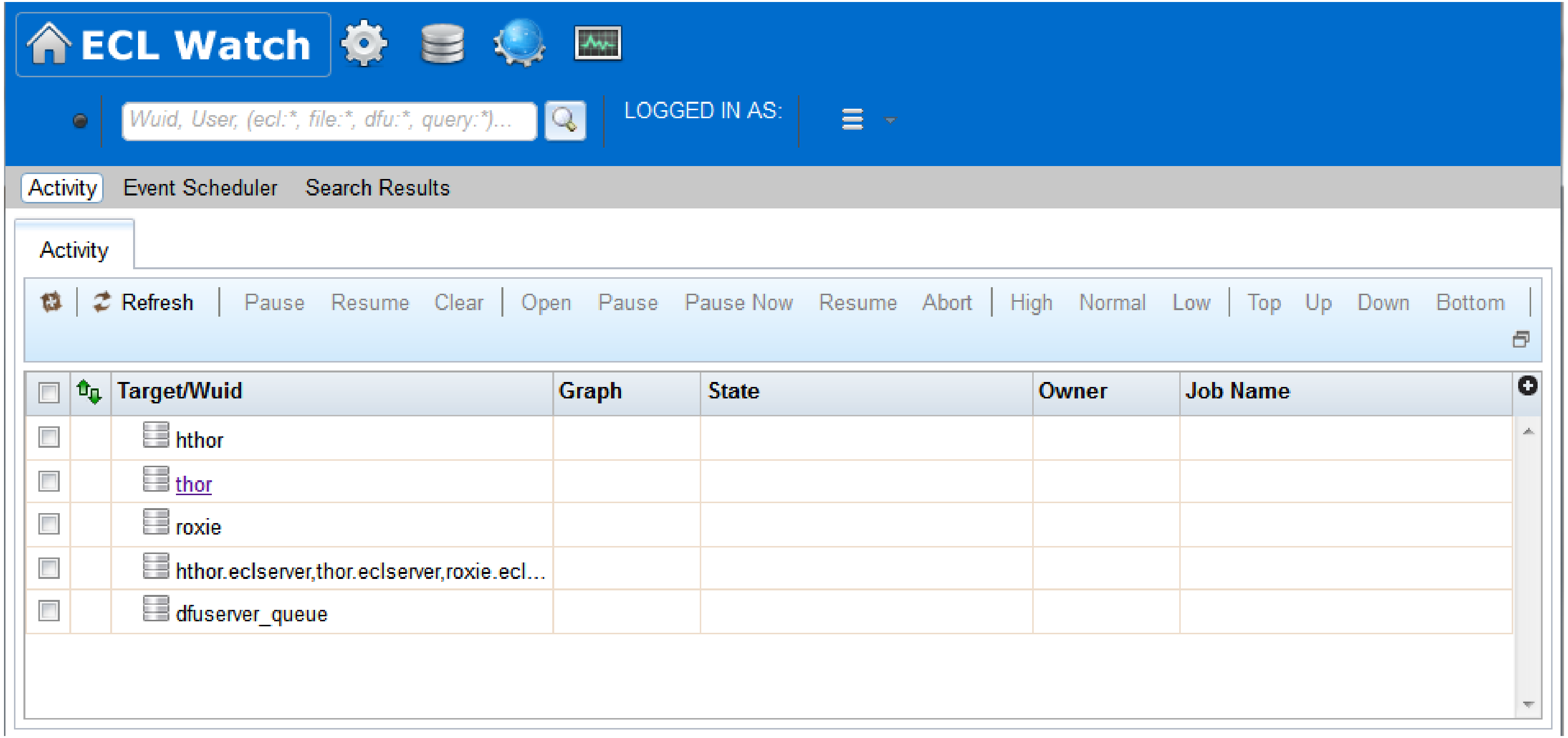
A full-featured GUI for ECL development providing access to the ECL repository and many of the ECL Watch capabilities.

Uses various ESP services via SOAP.

Provides the easiest way to create:

1. Queries into your data.
2. ECL Definitions to build your queries which:
   - Are created by coding an expression that defines how some calculation or record set derivation is to be done.
   - Once defined, can be used in succeeding ECL definitions.

The ECL Watch



□		Target/Wuid	Graph	State	Owner	Job Name
□		hthor				
□		thor				
□		roxie				
□		hthor.eclserver,thor.eclserver,roxie.ecl...				
□		dfuserver_queue				

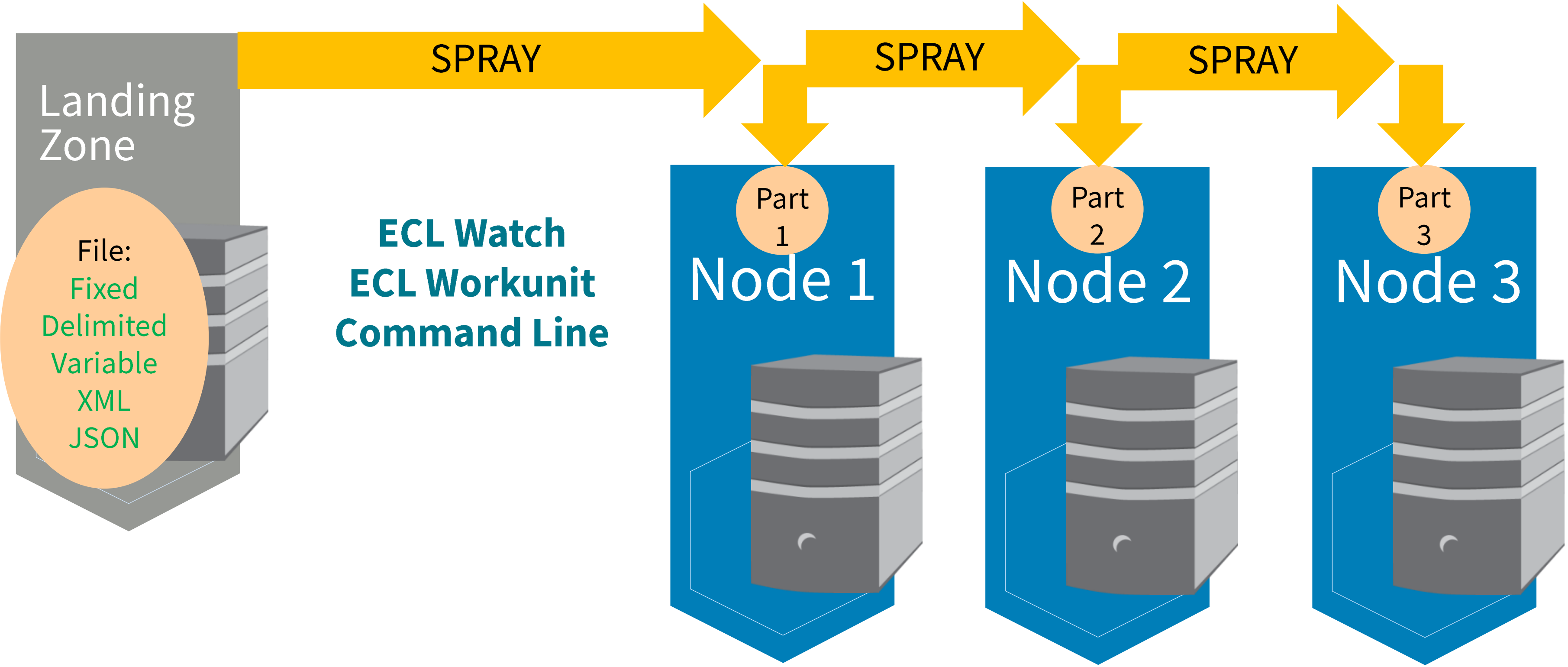
ECL Watch Features:

A web-based query execution, monitoring and file management interface. It can be accessed via ECL IDE or a web browser.

ECL Watch allows you to:

1. See information about active workunits.
2. Monitor cluster activity.
3. Browse through previously submitted WUs:
   See a visual representation of the data flow within the WU.
   Complete with statistics which are updated as the job progresses.
4. Search through files and see information including:
   Record counts and layouts.
   Sample records.
   The status of all system servers whether they are in clusters or not.
5. View log files.
6. Start and stop processes.

SPRAY Operation

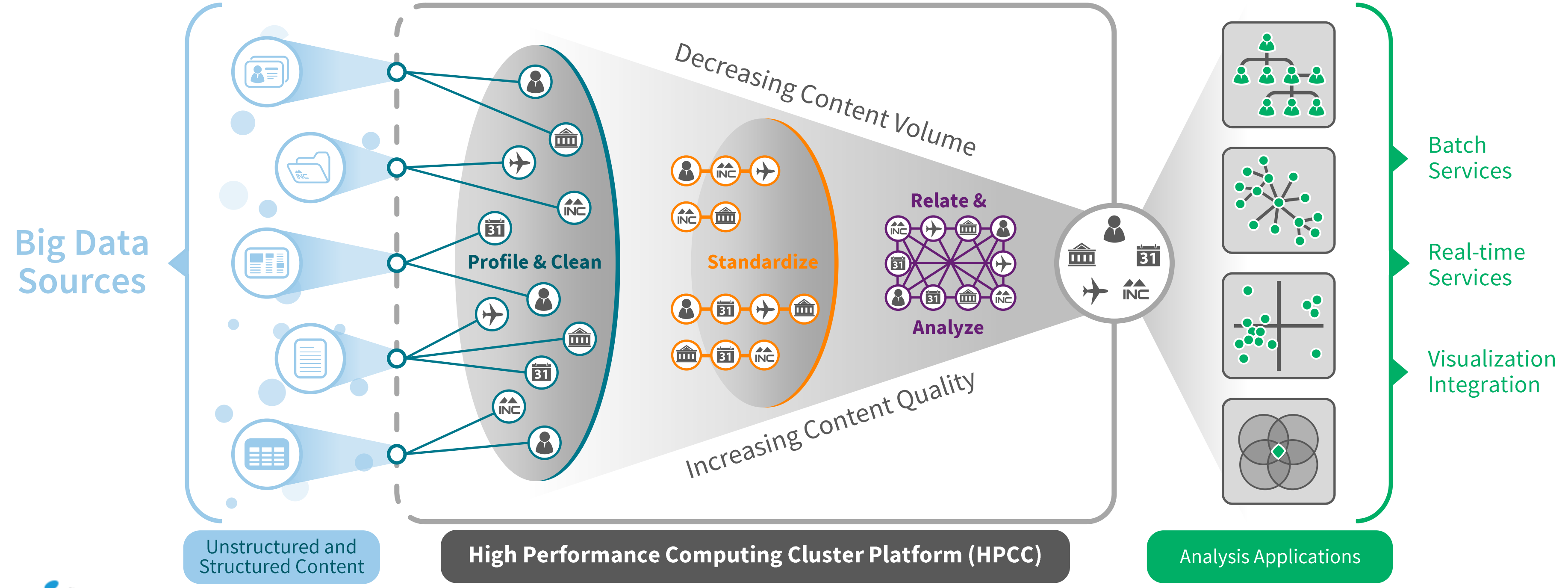


ECL Watch
ECL Workunit
Command Line

Treated as a single logical file...

HPCC Systems (Small to Big Data) ETL

Integrate > Profile > Clean > Standardize > Analyze > Deliver



Big Data Sources
Profile & Clean
Standardize
Relate &
Analyze
Decreasing Content Volume
Increasing Content Quality
Batch Services
Real-time Services
Visualization Integration
Unstructured and Structured Content
High Performance Computing Cluster Platform (HPCC)
Analysis Applications

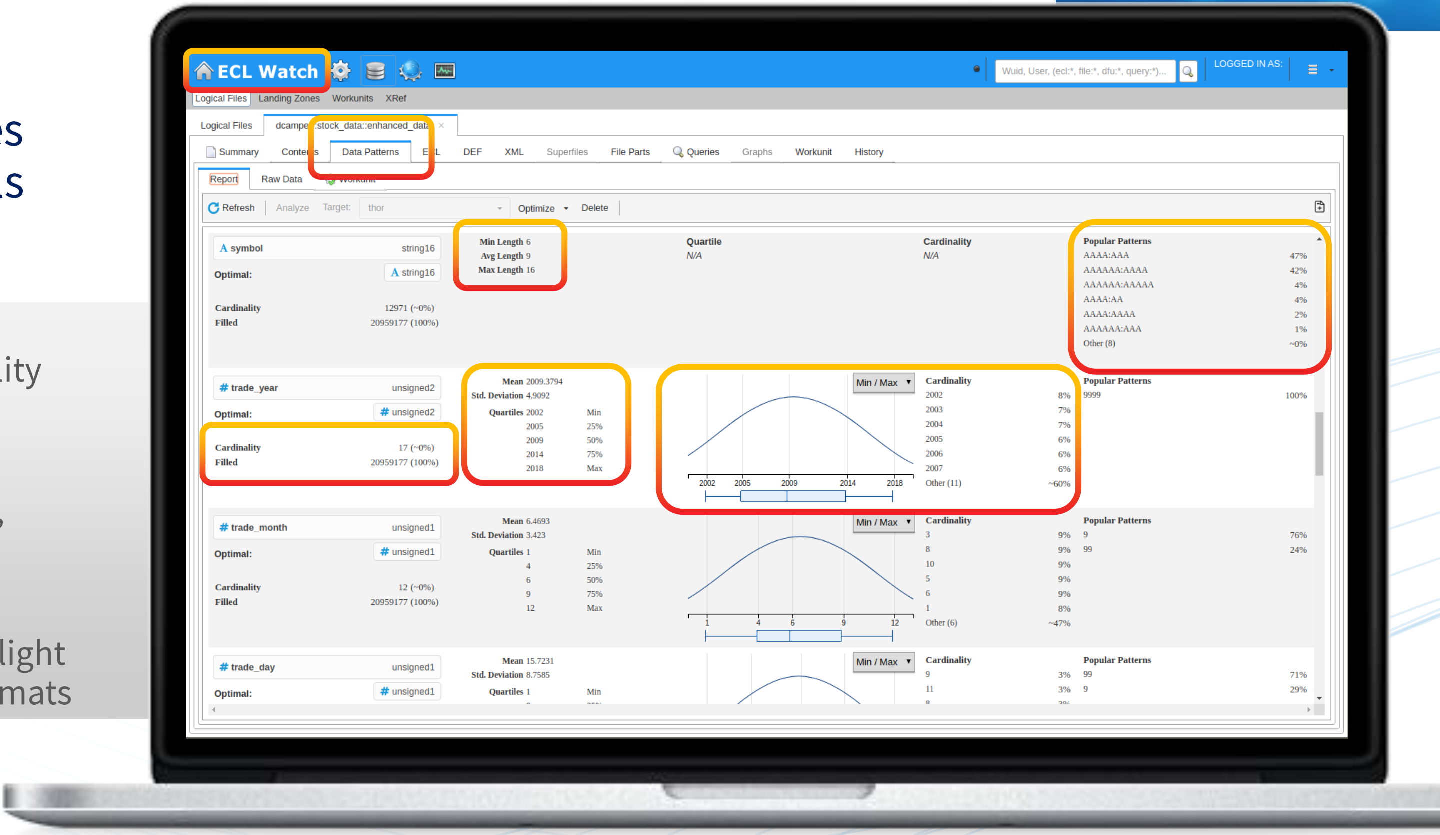
HPCC SYSTEMS®

Libraries and Plugins

Integrated Data Profiling

Built-in data profiling exposes field-level details

- Fill rates and cardinality details
- Numeric range detail, including quartiles
- Textual patterns highlight common and rate formats



A symbol string16
Optimal: A string16
Cardinality 12971 (~0%)
Filled 20959177 (100%)

Min Length 6
Avg Length 9
Max Length 16

Quartile N/A

Cardinality N/A

Popular Patterns	
AAAA:AAA	47%
AAAAAA:AAAA	42%
AAAAAA:AAAAA	4%
AAAA:AA	4%
AAAA:AAAA	2%
AAAAAA:AAA	1%
Other (8)	~0%

# trade_year unsigned2
Optimal: # unsigned2
Cardinality 17 (~0%)
Filled 20959177 (100%)

Mean 2009.3794
Std. Deviation 4.9092

Quartiles	
2002	Min
2005	25%
2009	50%
2014	75%
2018	Max

Cardinality	
2002	8%
2003	7%
2004	7%
2005	6%
2006	6%
2007	6%
Other (11)	~60%

Popular Patterns	
9999	100%

# trade_month unsigned1
Optimal: # unsigned1
Cardinality 12 (~0%)
Filled 20959177 (100%)

Mean 6.4693
Std. Deviation 3.423

Quartiles	
1	Min
4	25%
6	50%
9	75%
12	Max

Cardinality	
3	9%
8	9%
10	9%
5	9%
6	9%
1	8%
Other (6)	~47%

Popular Patterns	
9	76%
99	24%

# trade_day unsigned1
Optimal: # unsigned1

Mean 15.7231
Std. Deviation 8.7585
Quartiles 1 Min

Cardinality	
9	3%
11	3%

Popular Patterns	
99	71%
9	29%

It’s a Machine Learning World

Classical Machine Learning

Unsupervised

Clustering
- DBSCAN
- K-Means

Pattern Search
- Text Vectors
- Levenshtein Deletion Neighborhood

Dimension Reduction
- PCA

Supervised

Classification
- SVM
- Decision Trees
- Logistic Regression
- Classification Forest
- Latent Dirichlet Allocation (Topic Modeling)

Regression
- Linear Regression
- Regression Forest

Neural Nets & Deep Learning

- Autoencoders
- Convolutional Neural Networks
- Recurrent Neural Networks
- Perceptrons

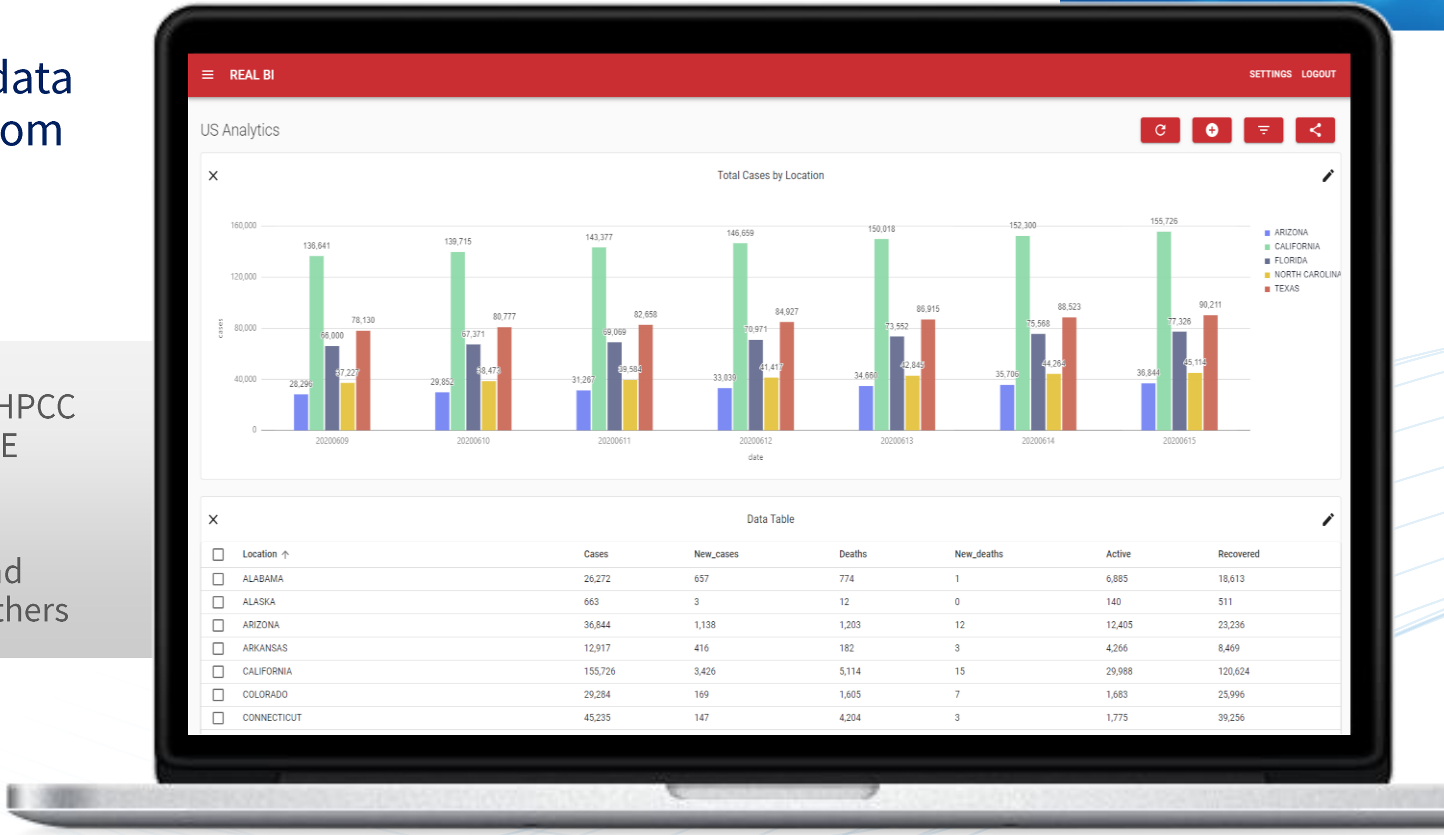
Ensemble Methods

- Random Forest
- Gradient Boosted Forest
- Gradient Boosted Trees

Visualization Bundle

Create custom data visualizations from HPCC clusters

- Chart live data from HPCC logical files and ROXIE queries
- Share dashboards and visualizations with others



□	Location ↑	Cases	New_cases	Deaths	New_deaths	Active	Recovered
□	ALABAMA	26,272	657	774	1	6,885	18,613
□	ALASKA	663	3	12	0	140	511
□	ARIZONA	36,844	1,138	1,203	12	12,405	23,236
□	ARKANSAS	12,917	416	182	3	4,266	8,469
□	CALIFORNIA	155,726	3,426	5,114	15	29,988	120,624
□	COLORADO	29,284	169	1,605	7	1,683	25,996
□	CONNECTICUT	45,235	147	4,204	3	1,775	39,256

HPCC Systems: Plugins

WsSQL

SPARK

JDBC/ODBC Driver

KAFKA

HPCC

PENTAHO

Couchbase

Tableau

SQS

Java API

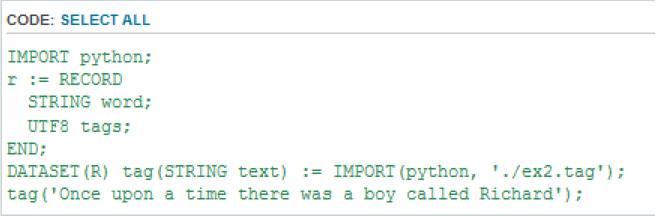
MEMCACHED

REDIS

Embedded Language

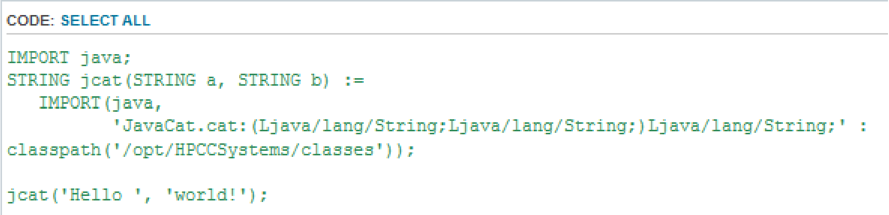
- C++
- R
- Python

- Java
- Cassandra
- SQL/SqLite



CODE: SELECT ALL

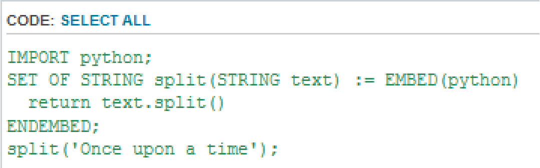
```
IMPORT python;
r := RECORD
  STRING word;
  UTF8 tags;
END;
DATASET(R) tag(STRING text) := IMPORT(python, './ex2.tag');
tag('Once upon a time there was a boy called Richard');
```



CODE: SELECT ALL

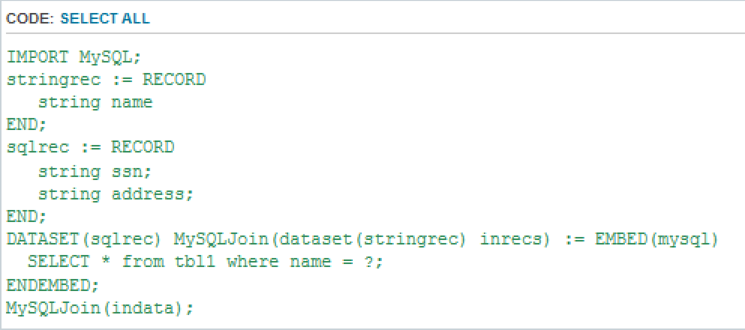
```
IMPORT java;
STRING jcat(STRING a, STRING b) :=
   IMPORT(java,
          'JavaCat.cat:(Ljava/lang/String;Ljava/lang/String;)Ljava/lang/String;' :
classpath('/opt/HPCCSystems/classes'));

jcat('Hello ', 'world!');
```



CODE: SELECT ALL

```
IMPORT python;
SET OF STRING split(STRING text) := EMBED(python)
  return text.split()
ENDEMBED;
split('Once upon a time');
```



CODE: SELECT ALL

```
IMPORT MySQL;
stringrec := RECORD
   string name
END;
sqlrec := RECORD
   string ssn;
   string address;
END;
DATASET(sqlrec) MySQLJoin(dataset(stringrec) inrecs) := EMBED(mysql)
  SELECT * from tbl1 where name = ?;
ENDEMBED;
MySQLJoin(indata);
```

Summary

Discover HPCC Systems, an end-to-end data lake management solution:

- A mature platform that has been heavily used in commercial applications for almost two decades
- Created by LexisNexis Risk Solutions and open source for nearly a decade now
- It is a powerful and versatile platform to work with and manipulate data as needed
- Makes it easier for your clients to query and find the data they need

Indeed, it is the Kit and Kaboodle for your Big Data Solutions!

Thank you!

Want to know more?

Portal:

https://hpccsystems.com

Free online training (138 classes and counting!):

https://learn.lexisnexis.com/hpcc

Free HPCC Clusters (try it out!):

https://play.hpccsystems.com:18010

Please email our training team:

training@hpccsystems.com

Join our Community

Help us make HPCC Systems better. Register on our community portal.

HPCC SYSTEMS®