HPCC SYSTEMS®

HPCC Systems:
Big Data NLP with HPCC Systems – A Development Ride from Spray to THOR to ROXIE

Bob Foreman – Senior Software Engineer/ECL Instructor

Twitter:
#HPCCMeetup

LexisNexis®

Risk Solutions

Welcome!

HPCC SYSTEMS®

- HPCC Systems Platform Overview
- Getting Started
  - Installations
  - Data Acquisition
  - Data Definition
- Parsing Semi-Structured Data
- Parsing Unstructured Data
- Post-Processing Parse Results
- Creating a Delivery Service

Primary Goals of this Meetup

HPCC SYSTEMS®

- Familiarity with the HPCC Systems Platform
- Familiarity with the ECL (Enterprise Control Language) Programming tools:
  - ECL IDE
  - ECL Watch
- Familiarity with Basic ECL Concepts and Syntax
- Familiarity with ECL's Parsing Technology

Where We're Going!

- Quick Demo:
  - Parsing Free Form Text (BWR_QuickDemo)
- Search Service
  - Search the Bible for matching words

Why Does HPCC Systems Exist?

HPCC SYSTEMS®

- 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.

The Result Of All That Development?

HPCC SYSTEMS®

HPCC Systems

A single, fully-integrated platform supporting the entire life cycle of Big Data product development:

- Raw Data Ingest – Thor
- Data Transformation to Product – Thor
- End-user Query Development – Thor
- End-user Query Delivery – Roxie

The Complete Big Data Value Chain

HPCC SYSTEMS®

Collection – collecting structured, unstructured and semi-structured data

Ingestion – consuming vast amounts of data including extraction, transforming and loading

Discovery & Cleansing - clean up, formatting and statistical analysis of the data

Integration – linking, indexing and data fusion

Analysis – statistics and machine learning

Delivery – querying, visualization, and redundancy, enterprise-class availability

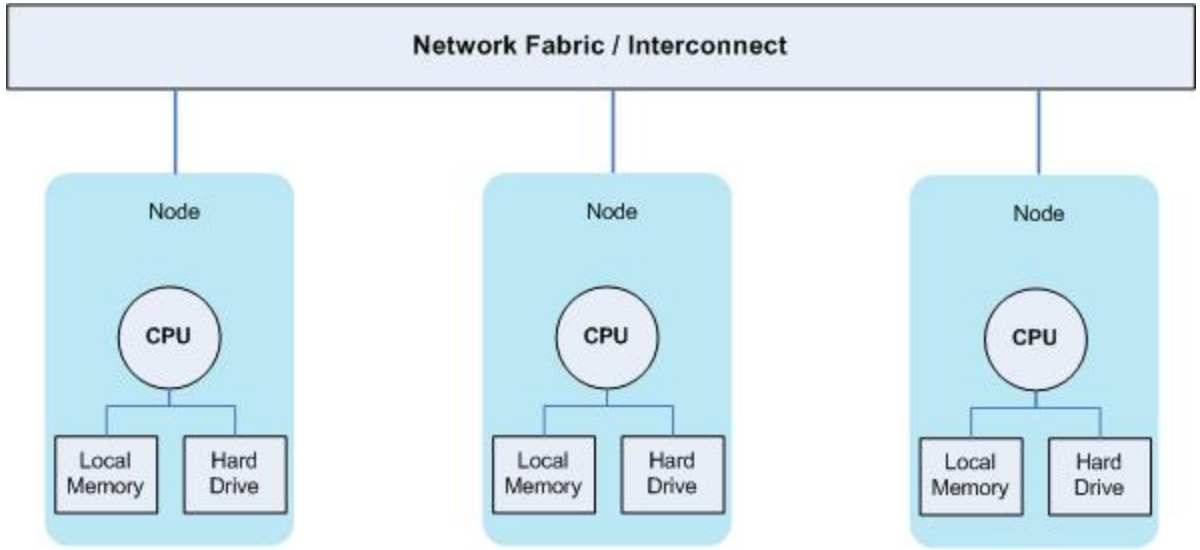
HPCC Systems Platform

There are two types of clusters in HPCC Systems:

- Data Refinery (THOR) – Processes every one of billions of records in order to create billions of "improved" records – runs one job at a time.
- Rapid Data Delivery Engine (ROXIE) – Searches quickly for a particular record or set of records – handles thousands of concurrent transactions per second.
- These are tightly coupled to the infrastructure that supports their operation, and the ECL programming language that defines the work done on them

HPCC Systems Hardware

- Clusters of commercial off-the-shelf components (COTS). Components are ideally homogeneous (all processing/disk storage components same) and the system is tightly coupled.
- Nodes are managed *en masse* instead of individually, which allows coordinated processing like global sorts (unlike Grid systems).



Thor Cluster

HPCC SYSTEMS®

- **Brute force:** Thor operates on massive amounts of data where datasets typically contain billions of records
- **Open Data Model:** The data model is defined by the user, not constrained by the limitations of a strict key-value paradigm
- **Scalable:** Horizontally linear scalability provides room to accommodate future data and performance growth
- **Truly parallel:** Datagraph Nodes can be processed in parallel as data seamlessly flows through them, effectively avoiding the well-known "long tail problem", resulting in higher and predictable performance.
- **Powerful optimizer:** The HPCC Systems optimizer ensures submitted ECL code executes at the maximum possible speed for the underlying hardware. Advanced techniques such as lazy execution and code reordering are thoroughly utilized to maximize performance

Roxie Cluster

HPCC SYSTEMS®

- **Low latency:** Data queries typically complete sub-second
- **Not a key-value store:** Roxie is not limited by the constraints of key-value data stores, allowing for complex queries, multi-key retrieval, fuzzy matching and more
- **Highly available:** Roxie operates in critical environments under the most rigorous service level requirements
- **Scalable:** Horizontally linear scalability provides room to accommodate future data and performance growth
- **Highly concurrent:** In a typical environment, thousands of concurrent clients can be simultaneously executing transactions on the same Roxie system
- **Redundant:** A shared-nothing architecture with no single point of failure provides extreme fault tolerance

HPCC Systems Platform

- **Batteries included:** All components create a consistent and homogeneous platform
- **Over 15 years of experience:** The HPCC Systems platform is the technology underpinning LexisNexis data offerings – its development began in 1999
- **Few moving parts:** HPCC Systems is an integrated solution extending across the entire data lifecycle, from data ingest and transformation to data delivery – no third party tools needed
- **Multiple data formats:** Supported out of the box, including fixed and variable length, delimited records, and XML
- **ECL inside:** One language to describe both: the data transformations in Thor and data delivery strategies in Roxie. Solutions to complex data problems are expressed easily and directly in terms of high level ECL primitives.
- **Consistent tools:** Thor and Roxie share the same set of tools, which provides consistency across the platform.

Data on HPCC Systems

- **Open Data Model:** The data model is defined by the user, as standard files and fields (tables and columns)
- **Simple:** Solutions to complex data problems can be expressed easily and directly in terms of high level ECL primitives
- **Implicitly parallel:** Data is always in distributed datasets whose parts are managed by the DFU, eliminating the need for programmers to manage the complexity of working with distributed datasets

Data on HPCC Systems

- Data is stored in ISAM Files
- Native support for:
  - Flat files, with fixed or variable-length records
  - CSV-type files (any delimiters may be used)
  - XML datasets
  - New JSON format support
- Each Record is always whole and complete on a single node
- A Record may have as many fields as needed
- Indexes are always LZW compressed and may contain “payload” fields in addition to search terms

What is ECL?

HPCC SYSTEMS®

- **Declarative programming language:**
  "Describes what needs to be done, not how to do it"
- **Powerful:** Unlike Java, high level primitives such as JOIN, TRANSFORM, PROJECT, SORT, DISTRIBUTE, MAP, etc. are available. Higher level code means fewer programmers and shorter time to deliver complete projects
- **Extensible:** As new definitions are created, they become primitives that other programmers can use
- **Implicitly parallel:** Parallelism is built into the underlying platform. The programmer need not be concerned with it
- **Maintainable:** A High level programming language, no side effects and definition encapsulation provide for more succinct, reliable and easier to troubleshoot code
- **Complete:** Unlike Pig and Hive, ECL provides for a complete programming paradigm.
- **Homogeneous:** One language to express data algorithms across the entire HPCC Systems platform, including data ETL and delivery.

Getting Started

- **Install:**

1. **Oracle's VirtualBox**:
   https://www.virtualbox.org/wiki/Downloads

2. **ECL IDE:**
   https://hpccsystems.com/download/developer-tools/ecl-ide

Getting Started

- **Run:**

1. Launch your VM player.

2. Import the HPCC Virtual Machine .ova file: http://hpccsystems.com/download/hpcc-vm-image

3. Note the IP next to the **IP Address:** prompt at the top of the VM.

   This IP address is the key to allowing the HPCC Systems client tools to access the environment.

Getting Started

- **Run:**

3. Open a browser and go to the ECL Watch page URL referenced in the paragraph following the IP Address.

   This web page is a key resource to have open while working with HPCC Systems. ECL Watch allows you to monitor the health of the environment, browse your datasets and workunits, and most importantly, to get data into and out of the HPCC Systems environment.

Getting Started

- **Run:**

4. Open the ECLIDE.exe application and:
   1. In the **Preferences** dialog, enter the previously noted IP (only) into the **Server** entry control.
   2. Go to the **Compiler** tab and press the **Add** button below the **ECL Folders** list.
   3. Select your target ECL folder(s) and press **OK**.
   4. Press **OK** to get back to the Login dialog,
   5. Press **OK** to login.

Acquiring Data

1. Task-switch to the **ECL Watch** page in your browser
2. Click the **Landing Zones** link in the **Files** section of the main ECL Watch menu
3. Press the **Upload** option and select the **actresses.list** file from your storage location.
4. In addition, select the **KJV.txt** file from your storage location.
5. Press the **Start** button from the Uploader dialog (big file may take awhile)

Acquiring Data

HPCC SYSTEMS®

1. Select the **actresses.list** file from your Landing Zone
2. Select the **Spray: Delimited** option to open the spray options.
3. Type **Meetup::<*initials*>** into the **Name Prefix** entry control
4. Modify the Target to **actresses**
5. Check the **Omit Separator** box and clear the **Quote** entry controls.
6. Press the **Spray** button
7. Monitor your DFU workunit to verify that the spray completed successfully!

Acquiring Data

HPCC SYSTEMS®

1. Deselect the **actresses.list** and select the **KJV.txt** file from your Landing Zone
2. Select the **Spray: Delimited** option to open the spray options.
3. Type **Meetup::<*initials*>** into the **Name Prefix** entry control
4. Modify the Target to **KJV**
5. Check the **Omit Separator** box and clear the **Quote** entry controls.
6. Press the **Spray** button
7. Monitor your DFU workunit to verify that the spray completed successfully!

Defining the Files

1. Task-switch to the **ECL IDE (overview of features)**
2. Open the **File_Actresses** ECL code file
3. Examine the code that defines the file
    1. The RECORD structure
    2. The DATASET declaration

Defining the Files

HPCC SYSTEMS®

1. Create a new **BWR_Actresses** ECL code file
2. Add the following code

```
IMPORT $;
OUTPUT($.File_Actresses.File,NAMED('Input_Data'));
```

3. Look at the defined data
   1. Select your Thor cluster as the **Target**
   2. Press the **Submit** button

Defining the Files

1. Open the **File_KJV** ECL code file
2. Examine the code that defines the file
    1. The RECORD structure
    2. The DATASET declaration
3. Open the **BWR_KJV** ECL code file
4. Look at the defined data
    1. Select your Thor cluster as the **Target**
    2. Press the **Submit** button

Overview of Natural Language Parsing

HPCC SYSTEMS®

Natural Language Parsing is accomplished in ECL by combining pattern definitions with an output RECORD structure specifically designed to receive the parsed values, then using the PARSE function to perform the operation.

Pattern definitions are used to detect "interesting" text within the data. Just as with all other ECL definitions, these patterns typically define specific parsing elements and may be combined to form more complex patterns, tokens, and rules.

The output RECORD structure (or TRANSFORM function) defines the format of the resulting recordset. It contains specific pattern matching functions that return the "interesting" text, its length or position.

The PARSE function implements the parsing operation. It returns a recordset that may then be post-processed as needed using standard ECL syntax, or simply output.

Parsing Semi-Structured Text

1. Open the **ParseActress** ECL code file
2. Examine the code
    1. PATTERN definitions
    2. The PARSE function
3. Look at the parsed result
    1. Select your Thor cluster as the **Target**
    2. Press the **Submit** button

Parsing Unstructured Text

1. Open the **ParseKJV** ECL code file
2. Examine the code
    1. The TABLE function
    2. The TRANSFORM structure
    3. The ROLLUP function
    4. The SORT function
3. Look at the parsed result
    1. Select your Thor cluster as the **Target**
    2. Press the **Submit** button

Post-Processing Parse Results

1. Open the **BibleNames** ECL code file
2. Examine the code
    1. The FUNCTION structure
    2. The IF function
    3. The SET function
    4. Inline DATASET definitions
    5. The DEDUP function (Training Examples)
3. Look at the result
    1. Select your Thor cluster as the **Target**
    2. Press the **Submit** button

More Post-Processing

1. Open the **BWR_MatchNames** ECL code file
2. Examine the code
    1. The JOIN function
3. Look at the result
    1. Select your Thor cluster as the **Target**
    2. Press the **Submit** button

And Even More Post-Processing

1. Open the **SecondDegree** ECL code file
2. Examine the code
    1. The TRIM function
3. Look at the result
    1. Select your Thor cluster as the **Target**
    2. Press the **Submit** button

Building a Search Service

HPCC SYSTEMS®

1. First, write the search and test in THOR.
2. Examine the code (**Inversion**)
    1. The INDEX statement
    2. The BUILD action
    3. The GRAPH function
3. Test the Inversion Code (**BWR_Inversion**)
4. Look at the result
    1. Select your Thor cluster as the **Target**
    2. Press the **Submit** button

Get the Search Ready for ROXIE

HPCC SYSTEMS®

1. Open the **SearchService** ECL code file
2. Examine the code
    1. The STORED workflow service – All you really need!
3. Publish the query to ROXIE
    1. Select your ROXIE cluster as the **Target**
    2. Press the **Compile** button
    3. Publish in the ECL Watch interface

That's All Folks!

And There's So Much More to Learn!!!

Thanks for attending!

Useful Links and References

- LexisNexis Open Source HPCC Systems Platform: http://hpccsystems.com
- Online Training: http://learn.lexisnexis.com/hpcc
- The HPCC Systems blog: http://hpccsystems.com/blog
- Wiki: https://wiki.hpccsystems.com/
- Our GitHub portal: https://github.com/hpcc-systems
- Community Forums: http://hpccsystems.com/bb
- Bible Search ECL Tutorial by David Alan Bayliss (Chief Data Scientist): http://www.dabhand.org/ECL/construct_a_simple_bible_search.htm

Deck - http://cdn.hpccsystems.com/presentations/meetup/meetup.pdf
Code - http://cdn.hpccsystems.com/presentations/meetup/code.zip
Data - http://cdn.hpccsystems.com/presentations/meetup/KJV.txt

Contact information:
Email: Robert.Foreman@lexisnexis.com

Questions?