HPCC Systems ® GSoC 2016 project preview

Lorraine Chapman
Consulting Business Analyst

January 2016

Proposed GSoC 2016 Ideas List

HPCC Systems GSoC/Interns Wiki

What happens when…

9th -19th Feb	Accepted organizations application period
29th Feb	Accepted organizations are announced
14th- 25th March	Students application period
25th April	Accepted students announced
25th Apr - 23rd May	Community Bonding Period
23rd May	Coding starts
22nd August	Coding ends

Find out more about HPCC Systems and GSoC

- Read various blogs about GSoC 2015 written by Lorraine Chapman: http://bit.ly/1OJUIBT

- GSoC 2015 completed projects details: http://bit.ly/1QvMEqO

- Keep in touch! Visit the GSoC Forum: http://bit.ly/1K6MTHs

- Want to know more about the projects? Visit our GSoC 2016 Ideas page: http://bit.ly/1Ummxl5

- Recommend the HPCC Systems GSoC/Intern Wiki to interested students: http://bit.ly/1UmqhDo

HPCC SYSTEMS®

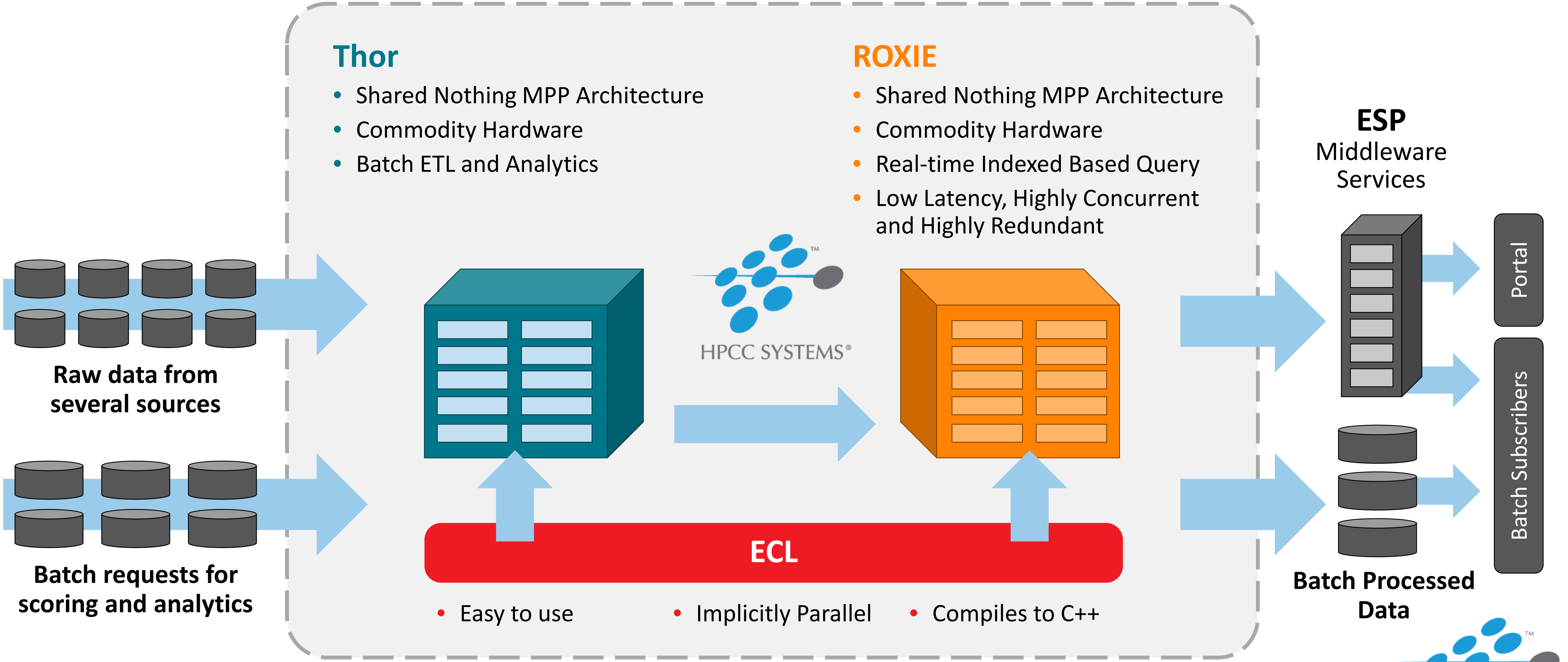
Highlights of features coming soon in HPCC Systems® 6.0.0

Lorraine Chapman
Consulting Business Analyst

January 2016

RELX Group

Data Flow Oriented Big Data Platform



Raw data from several sources

Batch requests for scoring and analytics

Thor

- Shared Nothing MPP Architecture
- Commodity Hardware
- Batch ETL and Analytics

ROXIE

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

HPCC SYSTEMS®

ECL

- Easy to use
- Implicitly Parallel
- Compiles to C++

ESP
Middleware
Services

Portal

Batch Subscribers

Batch Processed Data

HPCC SYSTEMS®

HPCC Systems 6.0.0 – What's the focus?

Performance ⊕

- **Virtual slave thor**
- Dali replacement for workunit storage
- Optimized merge sort for large numbers of cores
- Affinity support in Thor
- Parallel activity execution
- LZ4 compression for temporary files

Usability ⊕

- **Trace activity**
- Refresh boolean option on persist
- Improved ECL compiler error reporting
- DFUPLUS restore superfiles
- Security enhancements
- Init system improvements
- Ability to merge multiple package files
- NEW ESP service methods

New Features

- **HPCC Systems Visualization Framework**
- Kafka plugin
- RESTful roxie
- Security manager plugin support
- Quantile activity
- Option to bind queries to cores in ROXIE
- Dynamic ESDL support for writing web services in JAVA

Performance – Faster and even better

- **Optimized merge sort** - Faster execution of sorts and more efficient use of multiple core processors
- **Parallel activity execution** - Refactoring of the engines makes it easier to allow activities and sections of a graph to be executed in parallel
- **LZ4 compression for temporary files** – Implemented as the default in Thor for temp files (e.g. spills)

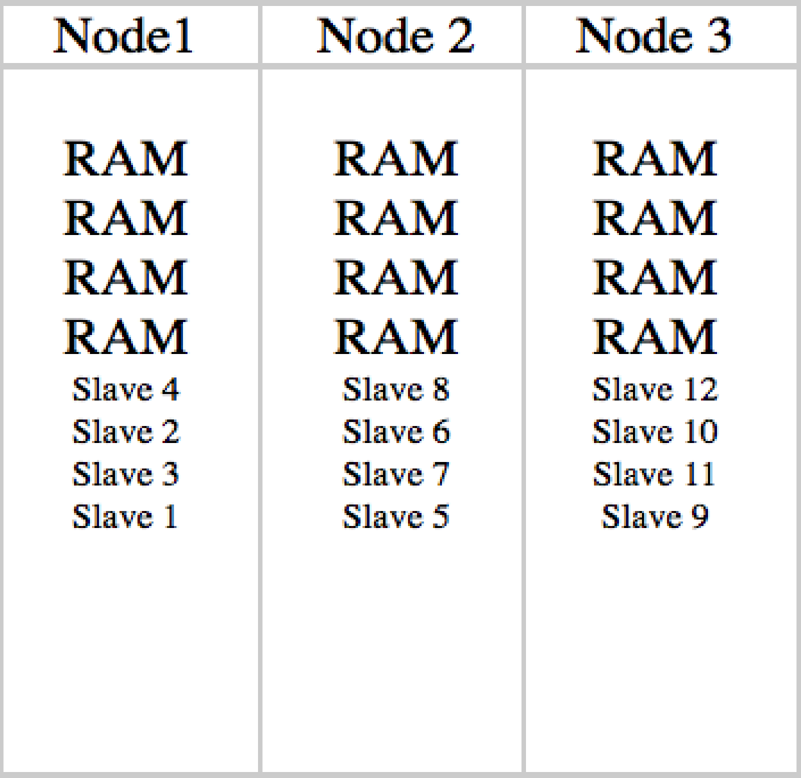
Performance – Faster and even better (cont’d)

- **Affinity support in Thor** - Improvement in overall performance because you can now bind each process to a single socket
- **Dali replacement for workunit storage** – The first release to contain an option to use a Cassandra database instead of DALI
- **Virtual slave Thor** - Thor clusters can be configured to take full advantages of the resources available per node

Thor - Cluster Configuration

Before – SlavesPerNode is set to N

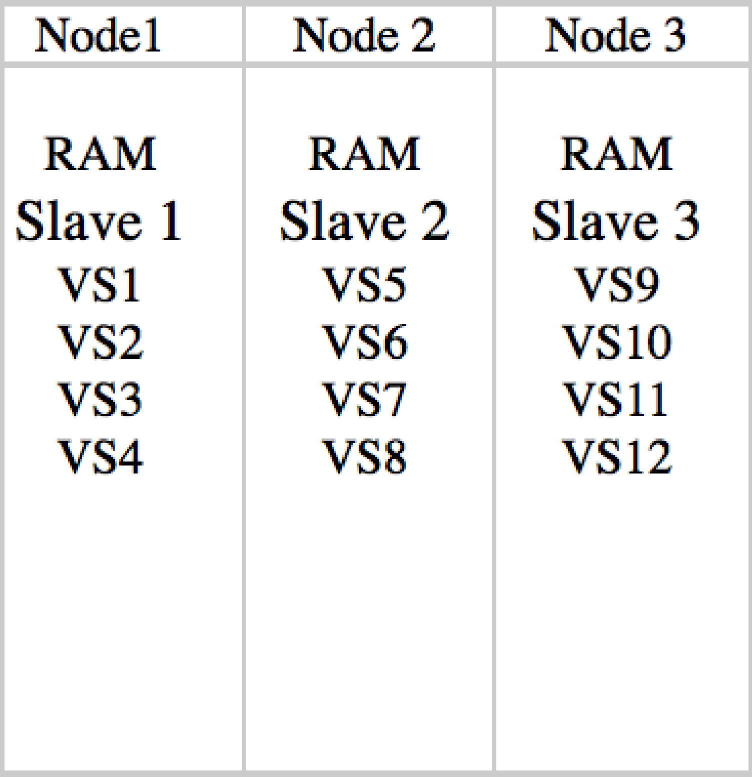
N independent slave processes per node



Node1	Node 2	Node 3
RAM RAM RAM RAM Slave 4 Slave 2 Slave 3 Slave 1	RAM RAM RAM RAM Slave 8 Slave 6 Slave 7 Slave 5	RAM RAM RAM RAM Slave 12 Slave 10 Slave 11 Slave 9

After – Virtual slave Thor

Virtual slaves created with a single slave process



Node1	Node 2	Node 3
RAM Slave 1 VS1 VS2 VS3 VS4	RAM Slave 2 VS5 VS6 VS7 VS8	RAM Slave 3 VS9 VS10 VS11 VS12

Virtual slave Thor – Sharing resources

- **Each virtual slave shares cached resources**
- **Slaves can request and share all available RAM**
- **Startup and management of the cluster is faster and simpler**
- **Access to all available memory is significant for some activities e.g. Smart/Lookup join**

Smart/Lookup join example

How does a lookup join work?

- **Streams local slave RHS dataset to all other slaves**
- **All slaves gather global RHS into 1 table**
- **Hash table based on the hard key match fields is built**
- **Slaves finish and the LHS is streamed/matched against the hash table producing joined results**

What is a ‘Smart’ join?

Lookup join evolved. Two things…

1. **If global RHS won’t fit in memory it is hash partitioned. The LHS is hash distributed and a local lookup join is performed.**

2. **If it cannot fit local RHS set into memory on <u>any given node</u>, both local datasets are gathered and sorted and a standard join is performed.**

Smart/Lookup join advantages on a virtual slave thor

✓ **N times as much memory available for RHS**

4 times as much in the Smart/lookup join example

✓ **Significantly less communication of row data**

Faster processing for larger RHS sets

Usability – Easier and even more efficient

- **Refresh boolean option on persist** – Can now read cached copies of data without requiring them to be rebuilt when out of date
- **Improved ECL compiler error reporting** – More specific errors and follow on errors kept to a minimum
- **Ability to merge multiple package files** – Smaller packages files can be added or removed individually and organized locally, only affecting those queries individuals/teams are responsible for

Usability – Easier and even more efficient
And even more…

- **DFUPLUS restore superfiles** – Restore all the files from a previous session including the ability to export and restore superfiles
- **Init system improvements** – Better checking to detect configgen failures including more specific error messages, improved startup for ROXIE and Thor and more.
- **New ESP service methods** –Temporarily change the ESP logging level when it is running (WSESPControl.SetLogging) and get the list of graphs you want (WsWorkunit.WUGrapQuery).
- **Trace Activity** – Add and control tracing which doesn't change the graph and can be left in place when not being used

The Trace Activity – What is it?

- TRACE provides a way of putting ‘tracepoints’ in your ECL code using the syntax:

  myds:=TRACE(ds [, traceOptions]);

- Some or all of the data going through that part of the graph is saved into a log file:

  TRACE: <name><fieldname>value</fieldname>…</name>

- The workunit debug value traceEnabled must be set.

- Request tracing on a deployed query in ROXIE by specifying traceEnabled=1 in the query XML.

TRACE Activity - Options

- Zero or more expressions which act as a filter
- KEEP (n) – How many rows will be traced
- SKIP (n) – n rows will be skipped before tracing starts
- SAMPLE (n) – Only every nth row is traced
- NAMED(string) – Name for the rows in the tracing

‘…much better than scattering OUTPUT statements throughout your code!’

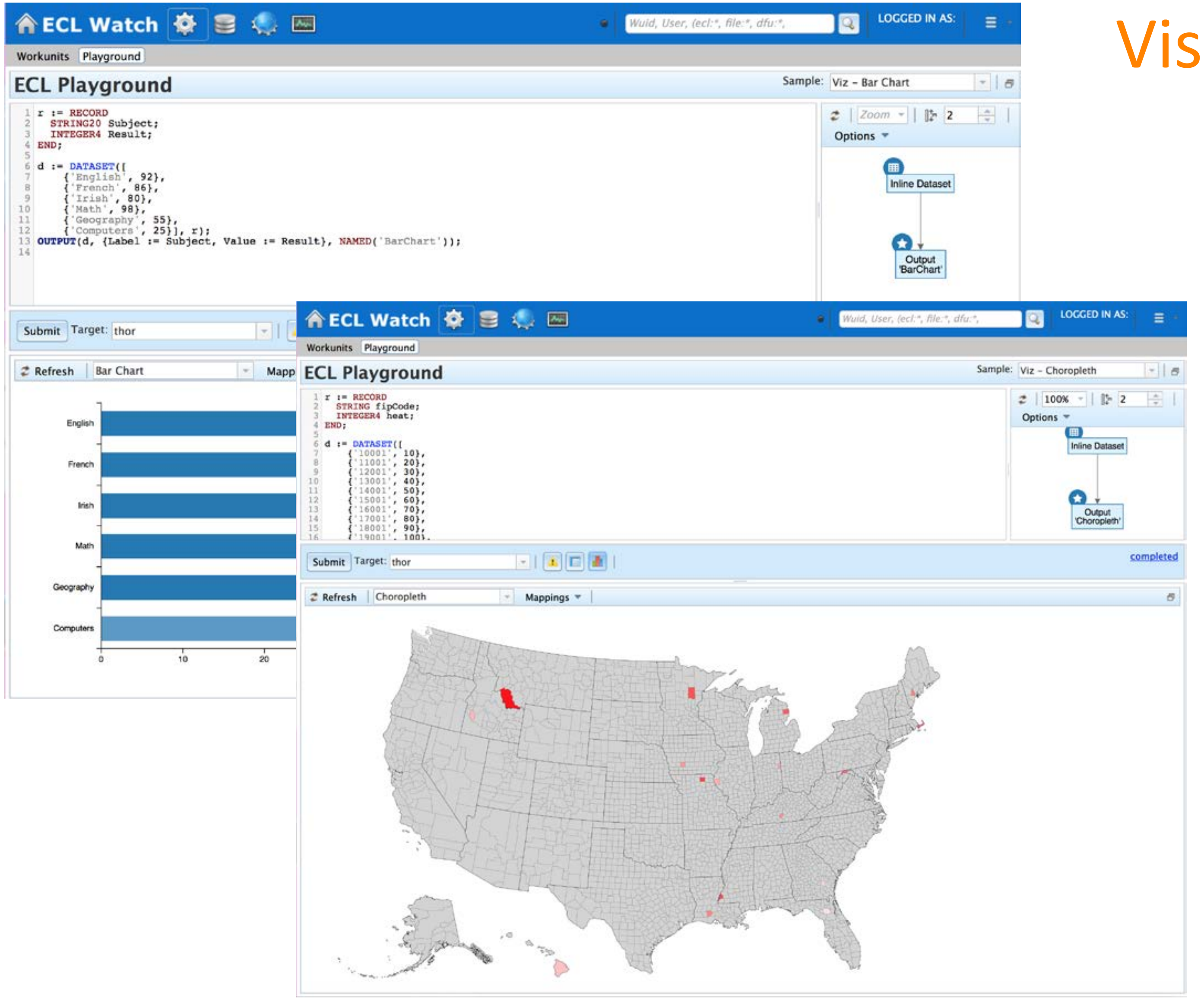
Gavin Halliday

New features – More cool and useful stuff

- **Kafka plugin** – Interface to Apache Kafka from ECL code
- **RESTful ROXIE** – Native ROXIE support for these additional REST access formats, SOAP, JSON, HTTP-GET, Form-UrlEncoding etc
- **Quantile activity** – Find the records that split a dataset into two equal sized blocks to locate the median, percentiles or split a dataset for distribution across the nodes in a system without performing a full sort

New features – More cool and useful stuff (cont’d)

- **Option to bind queries to cores in ROXIE** – Improve the performance of ROXIE using thread affinities to restrict a query’s threads to a subset of cores on a machine.
- **Dynamic ESDL support for writing web services in JAVA (Technical Preview)** – Configure an instance of Dynamic ESDL to run on ESP using the HPCC Systems Configuration Manager.
  Walkthrough: opt/HPCCSystems/examples/EsdlExample
- **HPCC Systems Visualization Framework** – a wrapper making it easier to put your HPCC Systems, or hand coded visualizations onto a web page.



```
r := RECORD
  STRING20 Subject;
  INTEGER4 Result;
END;

d := DATASET([
    {'English', 92},
    {'French', 86},
    {'Irish', 80},
    {'Math', 98},
    {'Geography', 55},
    {'Computers', 25}], r);
OUTPUT(d, {Label := Subject, Value := Result}, NAMED('BarChart'));
```

```
r := RECORD
  STRING fipCode;
  INTEGER4 heat;
END;

d := DATASET([
    {'10001', 10},
    {'11001', 20},
    {'12001', 30},
    {'13001', 40},
    {'14001', 50},
    {'15001', 60},
    {'16001', 70},
    {'17001', 80},
    {'18001', 90},
    {'19001', 100},
```

Visualizations in HPCC Systems

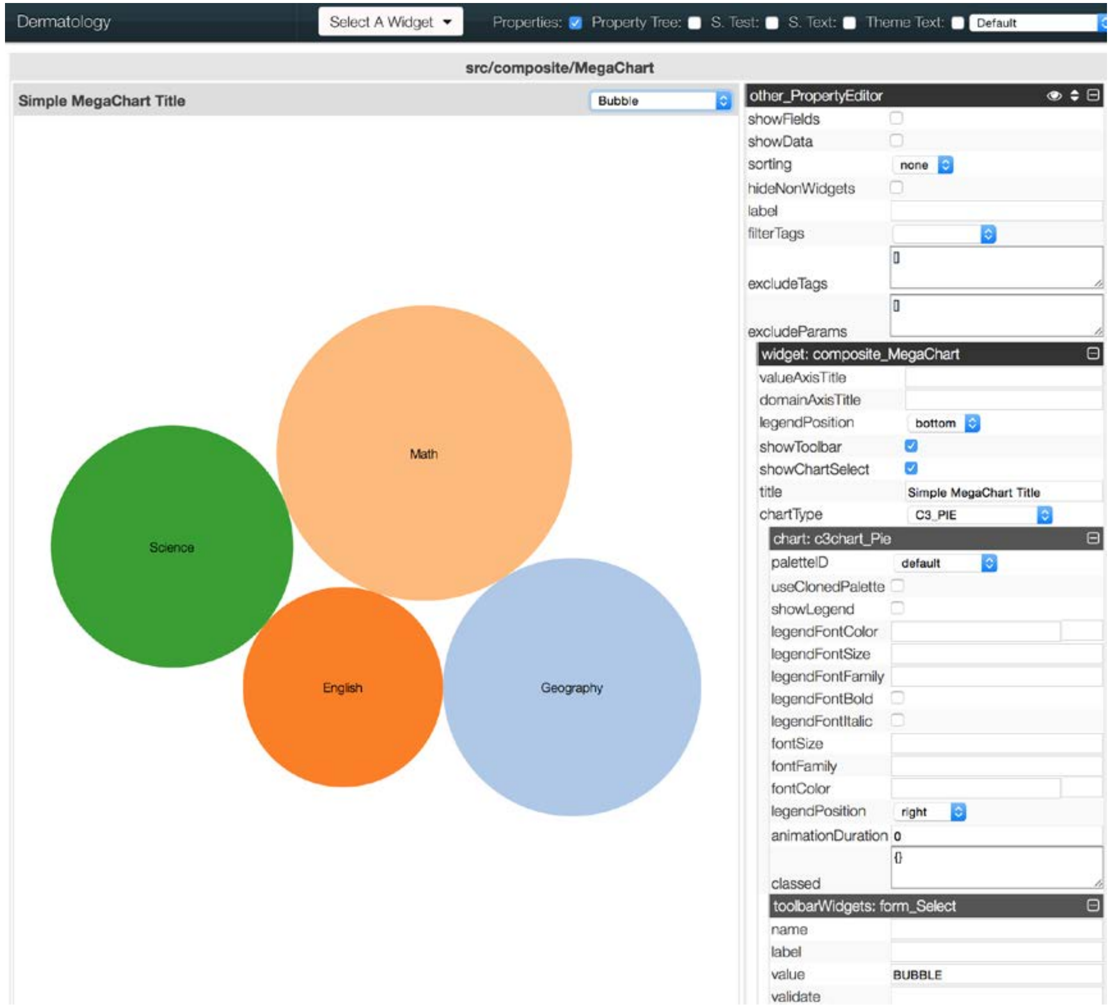
What's available now since 5.0.0…

- Visualizations for a number of chart/graph types including bar, scatter, pie, histogram etc.
- Examples available in the ECL Playground for bar and choropleth
- Include additional resources (e.g an index web page) in your ECL code via the manifest mechanism
- Helper functions to facilitate calling Roxie from inside web pages.

Visualizations Framework – Dermatology Page

If you want to go and play with any of the widgets:

- Look at the sources on github: https://github.com/hpcc-systems/Visualization
- Dermatology page for widget properties: http://rawgit.com/hpcc-systems/Visualization/master/demos/dermatology.html?src/map/Layered



showFields ☐
showData ☐
sorting none
hideNonWidgets ☐
label
filterTags
excludeTags []
excludeParams []

widget: composite_MegaChart

valueAxisTitle
domainAxisTitle
legendPosition bottom
showToolbar ☑
showChartSelect ☑
title Simple MegaChart Title
chartType C3_PIE

chart: c3chart_Pie

paletteID default
useClonedPalette ☐
showLegend ☐
legendFontColor
legendFontSize
legendFontFamily
legendFontBold ☐
legendFontItalic ☐
fontSize
fontFamily
fontColor
legendPosition right
animationDuration 0
classed {}

toolbarWidgets: form_Select

name
label
value BUBBLE
validate

Want to know more…

- Email Lorraine.Chapman@lexisnexis.com or post on the developer forum: http://bit.ly/23nLjYn
- HPCC Systems 6.0.0 Blogs – Beta 1: http://bit.ly/1WPkLKY and Beta 2: http://bit.ly/1VubyGI
- Quantile activity blogs: http://bit.ly/1nJzMlG
- Kafka plugin: https://github.com/hpcc-systems/HPCC-Platform/blob/master/plugins/kafka/README.md
- Dynamic ESDL docs: http://bit.ly/1njlXtA and example: /opt/HPCCSystems/examples/EsdlExample