Spraying Images into HPCC Clusters

Itauma Itauma

January 26, 2015

Contents

0.1 Working with Images in HPCC

In this report, I will show how Images are sprayed (loaded) into HPCC THOR cluster. The image dataset must be sprayed in BLOB format. THe dataset consist of multiple files. Easch image is naturally a file.

All images are placed in a directory. Use the BLOB spray option which will result in a datset on the cluster where each record is one of the image datasets.

The BLOB spray is described in the Programmer Guide document and in the Client Tools.

Typocally we use a prefix of both the name and length/ The record would look like:

```
ImageRecord := RECORD
STRING fileName;
DATA imageData;  //First 4 bytes contain the length of the image data
END;
```

The data string "imageData" is the file content.

0.2 Working with BLOBs

BLOB (Binary Large OBject) support in ECL begins with the DATA value type. This type may contain any tupe of data. making it perfect for housing BLOB data.

THere are essentially three issues around working with BLOB data:

1. How to get the data into the HPCC (Spraying).
2. How to work with the data, once it is in the HPCC. item How to get the data back out of the HPCC (Despraying).

0.2.1 Spraying BLOB Data

In the HPCCClientTools.pdf there is a chapter devoted to the DFUplu.exe program. DFUPlus is a command line distribute file utility management tool. It facilitate automation of data file spray, despray, and other common file handling tasks.

You can multiple spray all .jpg (or any image format) files in the directory to single logical file such as (BIGDATA::II::IMAGEDB) using the commandline or ECL Watch.

I will show you how to use ECL Watch to spray images.

0.2.2 Working with BLOB Data

Once you have sprayed the data into the HPCC you must define the RECORD structure and DATASET. THe following RECORD structure defines the result of the spray above:

```
imageRecord := RECORD
STRING filename;
DATA image;
END;
imageData := DATASET('~BIGDATA::II::IMAGEDB', imageRecord, FLAT);
```

The key to this structure is the use of variable-length STRING and DATA value types.