Market Requirements Document

Feature Name: Objectivity/HDFS Integration

Version: 0.1 Date Submitted: 03/06/2015 Completed By: Brian Clark

Description of the Problem

More and more 'big data' is being generated as the Internet of Things (IoT) takes hold. This data is typically stored in 'hdfs' clusters and processed by Map-Reduce algorithms. More and more uses of this data are being recognized, some of which include refining the data into more organized structures for more complex processing. Objectivity/DB has been used to store this refined data as objects and relationships in the database. Traditionally Objectivity/DB stores its database files in POSIX file systems. However there are now requirements to store the database files in the same file system as the raw data. One of the primary drivers here is to minimize the number of different file systems to manage from an operational perspective. Also to minimize the amount of data movement.

Background

Traditionally Objectivity/DB has stored its database files in POSIX file systems native to the host operating system. By its very nature most database accesses require high performance random i/o support. This is easily achieved using POSIX file systems and relatively small block sizes (Objectivity/DB uses between 1KB and 64KB database page sizes). Unix and Linux support POSIX file systems as standard. Windows is not so POSIX compliant but can be made to look so.

In the 'big data' world data access patterns tend to be different because of the volume of data and processing requirements. To support the data volumes and velocity the data is written once and appended to the end of the file. Processing tends to require reading large volumes of data into memory and processing there. This access is best supported using very large block sizes, typically 64MB and 128MB. Any new data is written out to new or appended to existing files.

'hdfs' is part of the Hadoop software stack. Apart from the open source Apache Hadoop implementation there are 3 major commercial implementations, Cloudera, HortonWorks, and MapR.

There are different tools to manage 'hdfs' clusters, specific to the actual Hadoop implementation.

Description of the Requested Feature

Objectivity/DB is required to support 'hdfs' as another file system, in addition to the supported standard POSIX file systems.

Part of an existing feature or does it require another feature, if so, which one?

• New feature, equivalent to supporting a new file system

How is this problem being solved now, and why isn't that acceptable?

• There is no current solution.

What languages must support this capability?

• Should be transparent to all supported language bindings.

Which platforms must be supported?

• Clients running on Linux, Windows and Mac OS X should be unaffected. Database servers should support this on platforms that support hdfs.

Do any competitors already have this feature?

• Yes, e.g. HBase, Titan.

Customers who require this feature

• Some existing customers e.g. CGG, new customers in the 'big data' space.

Revenue at risk, or which could be won

• Could lead to more early adopters.

When is this required?

• 'Objy 3.0'.

Additional Notes

1. Implementation notes:

- a. Initial implementation using oofs hooks and plugin. Need to add support for random i/o; and optimize block size for random i/o.
- b. Future implementation using ' multiple 'back-end' support.

2. Related Material

We will also need:

Marketing collateral, including a Press Release. Technical Publications. Release Note. Extra Quality Assurance Material.

3. Software requirements

a. hdfs implementation.

4. Hardware Requirements

a. Hdfs cluster