Market Requirements Document

Feature Name: Big Data Connection Platform

Version 2: 10/09/2012 Completed By: Leon Guzenda

Adds Graph Pattern Matching, Implementation Recommendations and more potential markets.

Version 1: 09/18/2012 Completed By: Leon Guzenda

Description of the Problem

Objectivity/DB and its derivatives (Objectivity/SQL++ etc.) provide classic database management system capabilities. InfiniteGraph is a Graph DBMS framework based on Objectivity/DB. Neither product directly addresses the primary needs of the "Big Data" market that has emerged over the past few years, primarily driven by the success of NoSQL technologies and Hadoop.

There are three primary NoSQL database technologies that are widely deployed in the Big Data world:

- BigTable clones that implement scalable, distributed tables, flexible consistency, and (sometimes) columnar access.
- Key-Value Stores
- Document Databases.

We and other Graph DBMS vendors are trying to position our products alongside the above technologies, with limited success to date. Here's how our existing products compare to the NoSQL technologies:

- Objectivity/SQL++ actually provides most of the placement and query features of BigTable, but lacks the flexible consistency and column-store capabilities.
- Objectivity/DB used to provide Keyed Objects, accessible via a primary key and placed in logical pages as if the pages were hash table buckets. Its scaleable hash collections aren't as scaleable as they should be.
- Objectivity/DB has been used to build multimedia and document storage applications, but it lacks any kind of standard document or media classes in its object class hierarchy. This is being partially rectified by the incorporation of JSON documents in InfiniteGraph Vertex (node) objects[1].

A key Big Data paradigm is that of parallel ("divide and conquer") processing, such as the Hadoop MapReduce framework. This is the prime reason that Oracle, IBM, HP, EMC and other vendors have incorporated Hadoop into their product lines, primarily in support of analytics (Business Intelligence). It would be possible to use MapReduce with either Objectivity/DB or InfiniteGraph, but we don't provide any tools or expertise to help users build applications using it.

---

[1] It may be worth considering providing JSON objects as a generic Objectivity/DB framework and have InfiniteGraph leverage the functionality.

Description of the Requested Feature

The Big Data Connection Platform is a package of new and existing features comprising:

a) **New Objectivity/DB features:**
   i. Document Objects, including JSON.
   ii. Replaceable File Systems – packaged, supported versions of OOFS for Lustre, GFS, ZFS and Hadoop HDFS (Tile Store only).
   iii. Co-operating Transactions – shared, named cache that can be written by one user, read by many and transferred to another thread/process.
   iv. Additional Event Notification features.
   v. An R Language API – to support analytics applications.
   vi. Workflow Management – starting with support for Hadoop MapReduce.
   vii. Security – a flexible framework, as defined in the Security "Goodies" Capsule.

b) **New InfiniteGraph features:**
   i. Packaged ETL tools, including import/export of standard document formats, such as JSON, and "easy pick" RDBMS interfacing.
   ii. Canned Visual Analytics Dialogs, integrated with the Workflow Management.
   iii. Graph Pattern Matching
   iv. Graph Event Notification – primarily for updating multiple visualizations as things change.
   v. Security (leveraging the feature above).

c) **A Tile Store** that supports extremely large nested arrays, chunks (BLOBs), key-value labels, streaming, versioning, provenance and security (leveraging the feature below). Its capabilities are defined in the Advanced Array Handling specification for MSC. It could integrate with Object File Systems, such as Lustre and DDN's WOZ.

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

- Prospects are choosing competing Big Data solutions, so we are losing revenue and traction in the market.
- We don't offer "jump start" capabilities that would make our products more easily accessible by builders of analytics applications.

What languages must support this capability?

- C++ (Essential)
- Java and .Net for C# (Soon)
- R [a new Objectivity/DB API]
- SQL++ (Highly desirable)

Which platforms must be supported?

- Linux, Mac OS X, Solaris and Windows, in that order.

Do any competitors already have this feature?

- The NoSQL, Graph DBMS and analytics platform vendors have some of the features. None of them has all of the features.

Customers who require this feature

- Big Data Analytics application/framework providers.
- Logistics applications (which could combine planning, simulation, quality and tracking information).
- Healthcare software providers, including telemedicine, robotic surgery and drug interaction applications.
- Scientific/Engineering applications, such as those being developed by MSC.
- Potential Partners/Customers:
  - Tom Sawyer and Tableau Software
  - Revolution Analytics
  - IxReveal, I2, SAP, kognitio and 1010data
  - Intel whamcloud (Lustre)
  - DDN WOZ
  - EMC, Panasas and Hitachi Data Systems
  - Oracle Big Data, IBM, Teradata, Lexus-Nexus and Bull Computer Group.
  - Column Store (Vertica, Sybase IQ, Infobright , ParAccelletc.) users.

Revenue at risk, or which could be won

- This capability could revive existing markets, particularly in the Intelligence and Defense space and open new markets (see above).

Related Material

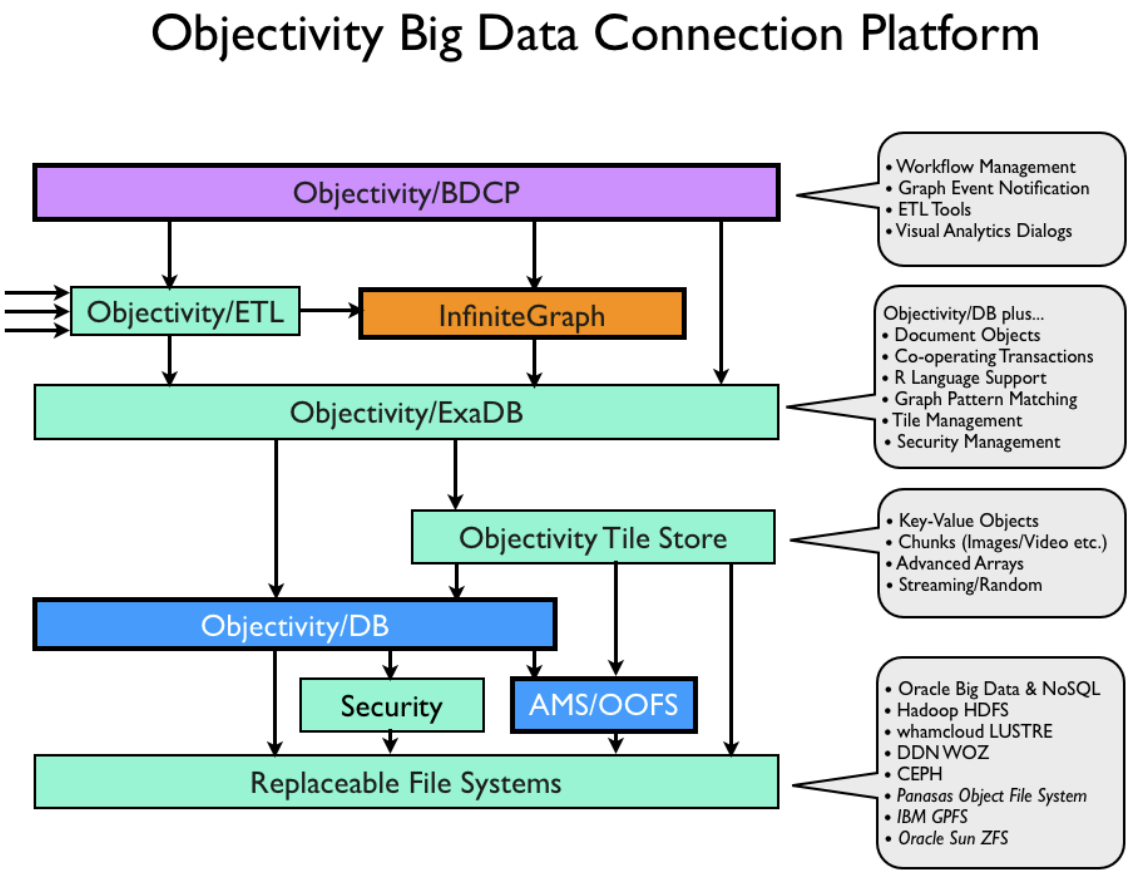
We will also need:

- New Quality Assurance material.
- Updated documentation, training and web based training.
- Collateral.

Implementation Recommendations

There are two primary goals:
1. Safe, low cost, quickest time to market and
2. Minimal disruption to existing products, especially the Objectivity/DB kernal.

The diagram below shows the various subsystems of the Big Data Connection Platform.



The recommended implementation sequence is:

a) Build the Objectivity Tile Store (OTS) to support Key-Value objects and "Chunks" that are equivalent to BLOBs in an RDBMS. Chunks can correspond to individual files, such as images and videos. The Tile Store should be able to use Oracle NoSQL (which can store Chunks as Key-Values) and Hadoop HDFS and/or Lustre initially. This can be achieved by modifying OOFS initially, but a local (client) binding will be needed eventually. The access to the various external file systems will be via the Replaceable File Systems layer, which is a packaging of existing OOFS functionality for specific file systems. The OTS can only be accessed via Objectivity/ExaDB, so a minimal implementation of its API is required.
b) Build the Advanced Array Handling needed by MSC. It will only be accessible via the Objectivity/ExaDB and OTS layers. Placement of tiles will be handled in a similar way to the standard Placement Manager, possibly, but not necessarily, by extensions to it. This feature set can be built in parallel with feature set a), possibly with its development accelerated with funding from MSC and others.
c) Build ETL tools for RDBMSs, Hadoop HDFS, regular files and Neo4J.
d) Add Document Objects (starting with JSON) to the Objectivity/ExaDB layer. This may simply be a matter of moving functionality from InfiniteGraph to the new layer.
e) Build workflow management tools, starting with MapReduce, into the Objectivity/ BDCP layer.
f) Add streaming to the OTS to support audio and video.
g) Add R Language support to the Objectivity/DB layer, supplemented with support for popular R tools and canned Visual Analytics Dialogs. This could be done in conjunction with, or even by, Revolution Analytics.
h) Add Graph Pattern Matching to the Objectivity/ExaDB layer (so that it can find patterns in InfiniteGraph and Objectivity/DB databases).
i) Add Graph Event Notification to the Objectivity/ExaDB layer. This could alert applications/users when new patterns are formed as a graph is modified.
j) Add Co-operating Transactions to the Objectivity/ExaDB layer. This will need support from the Objectivity/DB kernel for Objectivity/DB and InfiniteGraph databases, but not for the Tile Store, i.e. for MSC's requirements. It may be possible to find funding to accelerate this feature.
k) Add Security Management to the Objectivity/DB kernal (and hence InifiniteGraph) and also use it in Objectivity/ExaDB to protect data in OTS.
l) Add extra Visual Analytics Dialogs for popular tools, such as Revolution Analytics, which uses the R Language.
m) Extend the Workflow Management frameworkso that it does not depend on MapReduce and can deal with iterative cycles (such as design review processes and hill-climbing algorithms).