Market Requirements Document

Feature Name: **Objectivity/Grid – Release 10**

Version History:

Draft - 11/29/00 Version 1 – 8/6/01 Version 2 – 10/24/04
Version 3 – 01/14/05 Version 4 – 01/17/05 Version 5 – 02/09/05
Version 6 – 08/2/506

Author: Leon Guzenda

Introduction

This document identifies the subset of the grid enablement features that is required for Objectivity/DB Release 10. It first discusses the overall requirement, then lists the subset features in "Appendix A. Release 10 Grid-Enablement Subset." Readers familiar with this MRD may skip directly to Appendix A.

Background

Grid computing is being rapidly adopted. There are now many national and international projects underway to push the technology into production. IBM, Oracle and Sun have made grid technology a significant part of their business.

Peer-to-Peer [P2P] is a set of coordinated technologies that enable the direct exchange of data or services between computers. Peers can execute one or more tasks by transparently employing the power and storage of other peers. Client-Server computing is a small subset of the P2P paradigm. Sun's "The Network is the Computer" is one statement of the goals of P2P. "Networked" implies "Distributed" and vice versa for the purposes of this document.

A P2P environment must handle distributed processing (like CORBA or RMI), distributed data (such as the WWW and a good search engine) and distributed management of resources (such as OpenView). A grid generally uses diversely owned networks, resources and services transparently in order to accomplish individual or group computing goals. However, the tools being developed for administering grids are also proving useful for managing tightly clustered compute farms, such as those at SLAC and in the intelligence community.

As a simple example, a mobile user could pay a financial services provider to perform a detailed portfolio analysis using hundreds of collaborating peers over an interval of a few seconds. Or a large meteorological or air traffic control supercomputer could handle a sudden storm forecast overload by requesting help from thousands of trusted peers. It is

likely that a uniform charging mechanism will evolve for such services, maybe based on some combination of Teraflops, Gigabytes of RAM and Terabytes of persistent storage used. Others may choose to barter services based on some mutually agreed currency, e.g. $50K worth of my tape robot's storage for an equivalent share of your processor farm.

We are already seeing several kinds of early and mainstream adopters using grids for:

- Distributed sharing of rare resources
- Distributed servicing of heavy eCommerce loads
- Remote collaboration
- "Hero" applications – microclimatology, market forecasting etc.
- Efficient corporate resource management

Description of the Problem

Objectivity/DB, which uses a distributed client-server processing model, with individual machines capable of being both clients and servers, is an ideal fit for a grid environment. However, we need to make our server processors work with standard grid administration tools to make it easier for large sites to incorporate systems built on Objectivity/DB.

Interestingly, once we run cleanly within a uniform grid environment we should be able to add extra functionality, such as hot standby lock servers or AMS load balancing, without much effort. It should also be easier to adapt our product to suit a wider variety of environments, e.g. mobile clients and peers, or frequently disconnected peers.

Description of the Requested Feature

We need to enhance our product in the following areas:

- **Grid Platform**
  - Objectivity/DB should run on a distributed operating system, such as the Open Grid Forum's Open Grid Services Architecture [OGF/OGSA].
- **Security** – peers may only access data essential to their tasks.
  - Release 9 includes container-level files, making it easier to secure them and move them around filesystems.
  - Release 10 may also add page or object level encryption and authenticated links between clients and servers (AMS, Lock Servers and SQL++ Servers).

- **Loosely coupled communications:**

- Peers must be able to run across slow, high latency or unreliable networks, preferably using the Global Grid Forum [GGF] Open Grid Services Architecture [OGSA] protocols.
- A WAN version of AMS – reducing the latency and (optionally) increasing the packet size of client-AMS interactions
- Incorporation of extra hooks in the communications and OOFS layers. This will allow integration with proprietary or emerging standard protocols. We are also considering supporting TCP/Ipv6 at Release 10.
- Import/export file (or checkin/checkout) – to allow users to capture or re-create external files safely within transactions
  - XML export/import (done).
  - Transportable types – making it easier to add metadata to existing federations (this may be OK now with XML export/import).
  - High-speed data loader – the Release 10 version would rapidly load tabular data from conventional files into one or more containers and databases. Later versions might also have a streaming interface, e.g. for video/audio capture. The data loader should be able to use the rules defined by the Placement Manager.
- Post Release 10 - Container [or file] Replication – making it feasible to have tiered storage hierarchies (e.g. central, regional, in-store and personal).
- Post Release 10 - Deferred replication – allowing delayed replication across slow or unreliable networks.
- Post Release 10 - An ultra-lite client API that interfaces to a Database Server (basically, separating the current Language Interface and the Kernel). It should most probably be based on the GGF Data Access and Integration Services [DAIS].
- Post Release 10 - Downloadable product components - small & smart (automatically upgradeable).

- **Fault tolerance** – distributed transaction and process management, including disconnected peers.
  - Autonomous tools should make it possible to change schemas or catalogs without having to access all partitions.
  - Release 9.1 – JCA X/A Support, making it easier to co-ordinate updates to Objectivity/DB and other DBMSs.
  - Possibly post Release 10 - Partial backup/restore – removing the need to have to restore a complete federation.

- **Resource Management:**
  - Resource discovery – clients must be able to locate alternative resources on demand.

- Parallel query execution within, or on behalf of, a client process (addressed by the Release 10 Parallel Query MRD).
- Post Release 10 - Load balancing – an AMS or Lock Server should be able to share tasks with its peers.
- Post Release 10 - Resource charging – a peer must be able to advertise a cost for a particular kind of resource, monitor that resource and deliver a final "price". We may need to have our servers provide some new statistics.

The good news is that we only need to implement a few new features to have a viable grid product. The rest can follow as the market evolves. The most important features are:

- Integration with OGSA or Beowulf.
- The ability to run in a high latency or loosely coupled network.
- The ability to perform system tasks when some partitions are down.

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

- Enhances Objectivity/DB, Objectivity for Java and Objectivity/High Availability.
- Adds Objectivity/Grid

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

It isn't. Objectivity cannot be deployed in a high latency or low bandwidth network without adding an external messaging capability. Neither can it be deployed easily in an intermittently coupled network. These features are inherent in many of the commercial P2P and standard grid environments. Scientific and intelligence grids tend to use very high bandwidth networks, but they cannot use standard grid administration tools with our products.

What languages must support this capability?

- Java and C++

Which platforms must be supported?

- Tier 1 platforms at Release 10, then others on demand.

Do any competitors already have this feature?

- There are many distributed file systems and distributed resource management systems. Objectivity/DB ran successfully on SGI CXFS without any modification.
- IBM is grid-enabling most of its products, including DB2. IBM uses the GGF OGSA as their standard, heterogeneous platform.
- Oracle10g has extensive grid features, albeit mainly focused on administration tasks and their clustering capabilities, rather than heterogeneous environments.
- Platform Computing has announced their intent to integrate their products with Oracle10g.
- Microsoft is expected to make major grid announcements this year.
- The closest relevant competitive ODBMS feature is Versant's private database functionality, which isn't much better than our current (deprecated) "move container" feature. However, Versant is a centralized architecture.
- PointBase can run in a detached environment, but it uses a tiered client-server model. There is no parallelism.
- Birdstep is designed for P2P environments, but it is mainly aimed at mobile users.
- eXcelon has some load balancing in their "web database". It isn't a distributed DBMS. It would be hard to grid-enable it as it inherently relies on memory mapped files, which may not be supported by OGSA.

Customers who require this feature

- The VLDB market (intelligence, government, financial and scientific) has immediate prospects for this feature.
- Any large corporation using object-oriented languages and moving to a grid model is a "suspect" as the new applications probably won't be tightly coupled to legacy databases.
- Our government and scientific customers

Revenue at risk, or which could be won

New opportunities in the Federal and bioinformatics markets, both early adopters of the technology, could be at risk.

These companies could be VARs or partners:

- Platform Computing
- Entropia 2000 (Worldwide Projects List)
- Turbolinux - Enfuzion
- DataSynapse
- SARA Metacomputing
- Intel Peer-to-Peer Working Group:
    - Flycode (was AppleSoup)
    - Applied

- MetaComputing
- CenterSpan
- Distributed Science
- Dotcast
- Enfish Technology
- Engenia Software
- *Entropia*
- Groove Networks
- Hewlett Packard
- IBM
- Kalepa
- MangoSoft
- Popular Power
- Static
- United Devices
- Uprizer
- vtel

- Veridian – the ISV involved in the Information Power Grid.
- NetSolve (client to shared server)

Prospects:

- Global GRID Forum members. [**We should become a member**]
- European GRID Forum (eGrid) – this is also a partner in the Global GRID Forum *[I have a meeting with the UK members this week]*
- Global Information Grid (GIG) In 1999, the U.S. Department of Defense launched a major initiative for grid computing. It will be the framework and services (called Network-centric Enterprise Services, or NCES) into which all DoD systems are to fit to optimize integration, interoperation, and data/resource sharing. Its definition and implementation plan is still fuzzy in a lot of places, but DoD appears committed, so all of the systems designed for them must address how they will work within the GIG. It is likely that existing and prospective DoD clients will be asking how Objectivity will work on the GIG.
  - http://www.dtic.mil/whs/directives/corres/pdf/d81001_091902/d81001p.pdf
  - http://www.defenselink.mil/nii/org/cio/gpmlinks.html
  - http://www.disa.mil/pao/fs/gigbe3.html
  - http://www.disa.mil/pao/fs/nces3.html
- Information Power Grid - NASA (NAS Systems Division) - *[Note: We will have a meeting with the Project Manager this month or early January.]*
- National Computational Science Alliance (National Technology Grid)– HQ at University of Illinois – has 50 partners
- NPACI - National Partnership for Advanced Computational Infrastructure (National Technology Grid)
  - San Diego Supercomputing Center - Reagan Moore, Associate Director of SDSC's Enabling Technologies group is the Principal Investigator.
  - Naval Command, Control and Ocean Surveillance Center in San Diego
  - Entropia is involved
  - Distributed Object Computation Testbed (DOCT) is building its own ORDBMS interface to HPSS. We should approach them.
- HTMT – A Hybrid Technology Multithreaded Architecture for Petaflop Computing - Company It will use superconducting logic and holographic optical storage:
  - NASA, JPL and Argonne National Labs
  - Caltech, Princeton, University of Delaware, SUNY, University of Notre Dame, Argonne
  - Tera Computer
- The Metasystems Thrust
  - AppLes
  - Legion
  - Globus Project (Alliance, NPACI)
  - Network Weather Project
- Discom2 (Sandia, Lawrence Livermore and Los Alamos National Laboratories)
- Seti@home [Donate, minus Services. This would get Objectivity onto hundreds of thousands of machines.]
  - DataSynapse – "Building the World's Largest Peer-to-Peer network" – Uses compute time for non-profit users and pays the providers in Flooz (an eCurrency) [We could donate software and automatically give the Flooz to charity for a tax deduction. It has tens of thousands of users.]

- MetaNEOS (Argonne National Laboratory) – The MW API uses the Parallel Virtual Machine (PVM) and it can be downloaded from here
- Milan (New York University) – Metacomputing in Large Asynchronous Networks – Uses only COTS components. [Donate]
    - Calypso is a part of Milan and is a collaboration between NYU and Arizona State University. It is C++ and just four extra keywords! Calypso runs on Solaris, Linux, and Windows NT and is freely distributed. [Donate]

When is this required?

- Release 10

Additional Notes

We will also need:

- Marketing collateral
- Qualification questions/notes for ISPs, VARs and projects
- Sales training material
- A list of publications and industry events to be approached/attended
- A Grid seminar
- Membership in the GRID Forum - (Silver sponsorship? <$10k) - Remote Data Access Working Group (Data-WG)
- A Zero Defect Product Philosophy (being addressed by the Quality Assurance Working Group).
- Zero Learning Time Philosophy (addressed by Release 8.0, particularly Assist and the samples and Release 9 Web Based Training).

Useful Technologies:

- **Security :**
    - SSH (Secure Shell), Grid Security Infrastructure (GSI) or Kerberos)
- **Communications with remote resources:**
    - Scheduling via LSF (Load Sharing Facility) , PBS (Portable Batch System), or NQE (Network Queuing Environment), Codine, GRAM & NQS ,LoadLeveler.

  - The Resource Specification Language (RSL) is another Globus component
  - Corba Naming Service, CORBA Trader Service
  - Legion common context space
  - Grid Information Service (GIS), a part of Globus which uses a LDAP (Lightweight Directory Access Protocol) server for resource discovery

- **Communication and data access:**
  - OGF – Grid RPC etc. http://www.ggf.org/ggf_areasgrps_overview.htm
  - General Inter ORB Protocol (GIOP),
  - Nexus and the MPICH-G library
  - Talarian *(we've had an initial contact)*

- **Fault tolerance:**
  - FTO/DRO + Legion Single Program Multiple Data
  - Globus Heart Beat Monitor (HBM)

- **Loose Coupling (+checkout/in):**
  - Must work on a High Performance Network such as Abilene or vBNS.
  - Need to run over CORBA, Legion and Globus (both designed for high performance computing users)
  - SNMP & Multirouter traffic grapher (MRTG)
  - Internet 2 Distributed Storage Infrastructure (I2-DSI) service

Useful Resources:

- Beginner's Guide to Network-Distributed Resource Usage
- Open Grid Forum - http://ogf.org/ - This will gradually merge in all of the GGF material.
- Global Grid Forum - http://www.ggf.org/

APPENDIX A. RELEASE 10 GRID-ENABLEMENT SUBSET

Virtualization

Virtualization Step 1 – Use Internet protocols to locate a bootfile.

Requirement 1: It should be possible to specify a read-only URL for the bootfile path, for example:

http://one.objy.com/federations/demos/telecomDemoFD.boot

Notes:

a) Moving a boot file to the Internet location can be a manual step. The tools which create and change the bootfile do not need to recognize the URL. They should continue to work with the current syntax. This avoids the need to create web based interfaces for this release.
b) On Windows, it should be possible to use a Network Place, such as \\MainPC\SharedDocs\telecomDemoFD.boot, as a URL. In this case, it may be desirable to allow tools to create and edit the bootfile.

Virtualization Step 2 – Use OGSA protocols to locate the bootfile.

The Open Grid Forum [OGF] Open Grid Services Architecture [OGSA] has two potential mechanisms for virtualizing access to remote files:

- The Resource Name Service.
- The Grid File System (WG-GFS).

For Release 10, we should make it possible to locate a bootfile using the Resource Name Service. This may be as simple as implementing Step 1 and then having the user register the URL of the bootfile with the Resource Name Service. However, tools have to recognize the resource name, perhaps by introducing syntax similar to "RNS:telecomDemoFD" or a new switch "-resourcename".

Virtualization Step 3 – Use OGSA protocols for client-server communications.

Objectivity/DB clients currently communicate with servers using TCP/IPv4. We will allow TCP/IPv6 at Release 10. The OGSA specifies a Grid RPC, which will work with IPv4 and IPv6. It should be possible for clients to use Grid RPC to communicate with servers. This will require the registration of the server type (AMS, lockserver, query serveretc.) and server location (virtual host name) with the OGSA Resource Name

Service. The Resource Name Service takes care of the mapping between virtual host name and physical host name, which is what we currently catalog.

Standards Compliance

Standardization Step 1 – Certify Objectivity/DB at IBM Grid Compliance Level 6

Objectivity/DB 9.0 was certified to run at IBM Grid Compliance Level 3, i.e. the servers can be located on known physical locations on a grid and applications can be started anywhere, using grid scheduling middleware. The clients must have access to a bootfile at a specified physical location, or locally.

IBM Grid Compliance Levels 1 to 3 apply to batch programs. Levels 3 to 6 apply to Service Oriented Architecture (ideally WebSphere) enabled applications. Objectivity/DB 9.3 is being certified with WebSphere 6.0, which is grid-enabled. So, it is logical to certify Objectivity/DB 9.3 and/or 10 with WebSphere 6.0 and validate that the grid paradigm works correctly.

At minimum, it should be possible for a lightweight client to route a request via WebSphere to an Objectivity/DB application (service) that accesses a federation using a local bootfile and physical server locations (hostnames).

Ideally, a Release 10 “virtualized” build should be used, removing the need to know the physical locations of the bootfile and servers.

Standardization Step 2 – Other Grid Protocols

Certify Objectivity/DB with the Sun Grid or the Argonne National Laboratory Access open grid environment. This should only be done if we can locate a customer or prospect interested in beta testing the product in one of these environments.

Exclusions

1. Note that, unless it is a natural by-product of virtualization steps 1 to 3, there is no Release 10 requirement to use virtualized file locations for any file other than the bootfile.

2. Communications in a grid may be low bandwidth and high latency. It will be acceptable for Release 10 to require a high bandwidth, low latency communications infrastructure.

Quality Assurance

1. It will be necessary to establish a grid testing platform for Release 10, using OGSA.

2. It would be desirable to modify the QA test harness to exploit the grid environment.