

Proposal for an OODBMS-based Event Store

David R. Quarrie
Software Technologies and Applications Group
Information and Computing Sciences Division
Lawrence Berkeley National Laboratory

LBL - MS 50B-3238
1 Cyclotron Road
Berkeley, CA 94720
(510) 486-4868
DRQuarrie@LBL.Gov

David Quarrie - Proposal for a OODBMS-based Event Store

Overview

- Objectivity/DB Overview
- Event Collections
- Event Structure
- Regional and Institution Centers
- Plans
- Status & Issues

David Quarrie - Proposal for a OODBMS-based Event Store

Objectivity/DB Overview

- Client/Server Object Database Management System
 - Chosen by RD45 as “best candidate” OODBMS
- Fully distributed
 - Heterogeneous
 - Multiple servers & clients
- C++, C & Smalltalk APIs
- ANSI-standard SQL interface
 - Possible mechanism for migration of Oracle data
- Page-based server
 - Once object is in local cache it is directly accessible
 - Object localisation allows prefetching of other objects
 - Cross-transaction caching for reused objects
- Almost conforms with ODMG-93
 - Object Database Management Group

David Quarrie - Proposal for a OODBMS-based Event Store

Objectivity/DB Object References

- Smart object references
 - Instead of pointers
 - Have same semantics as pointers
 - Smart reference has its own methods
 - Transparent caching/decaching

```
d_Ref<T>          // ODMG-93 (Objectivity/DB v4.0)
ooRef(T)         // Objectivity/DB v3.8
```
- Smart handle
 - Like a reference but “locks” object (cannot be decached)
 - Only transient (cannot appear in a persistent object)

```
ooHandle(T)
```
- Use of C++ pointers not supported
 - They do have a tool to convert T* to ooRef(T) but not fool-proof

David Quarrie - Proposal for a OODBMS-based Event Store

Objectivity/DB Hierarchy

- Federated Database
 - One per project (contains schema, catalog of databases, etc.)
 - A single application can only access a single federated database
- Database
 - Corresponds to a Unix file
 - Up to 32k per federated database
 - Maximum database size is determined by file system
- Container
 - “Clustering” of objects within a database
 - Granularity for locking (transactions)
 - Supports multiple indexes for rapid searches
 - Up to 32k per database
- Page
 - Normally 8192 bytes
 - Unit of granularity for object access
 - Prefetching of small objects

David Quarrie - Proposal for a OODBMS-based Event Store

Objectivity/DB Database concepts

- Concurrency
 - Locking
 - Read, update
 - Multiple levels of granularity with container being lowest level
 - Based on lock server process
 - Transactions
 - Checkpointing
 - Long transactions
 - Multiple readers & one writer (MROW) per container
- Object-level versioning
 - Create separate versions & create genealogy
 - Access latest or specific version by default
- Schema evolution
 - All objects of a class (across whole federation)
 - All objects within a container or database
 - As needed (when object is accessed)

David Quarrie - Proposal for a OODBMS-based Event Store

Objectivity/DB Object Modelling

- Relationships
 - One to one
 - One to many
 - Many to one
 - Many to many
- Bidirectional or Unidirectional
 - Bidirectional support referential integrity
 - No dangling references
 - Unidirectional take less space

David Quarrie - Proposal for a OODBMS-based Event Store

Objectivity/DB C++

- Data Definition Language (DDL)
 - DDL processor converts a .ddl file into several C++ source & header files
 - DDL is not ODMG-93 ODL but there is a converter
- Defining Persistent objects
 - Inheritance from persistent base class (ooObj or d_Object)

```
class MyClass : public ooObj {
    ...
};
```

- Relationships

```
class Library : public ooObj {
    ...
    ooRef(Book) allBooks[] <-> fromLibrary:// One to many
    ...
};
class Book : public ooObj {
    ...
    ooRef(Library) fromLibrary <-> allBooks[]:// Many to one
    ...
};
```

David Quarrie - Proposal for a OODBMS-based Event Store

Level 3 & PASS1 Reconstruction

- Raw Data spooled to disk from Level 3 farm
 - Several Level 3 nodes (~10)
 - Even if technically feasible for a single node to do everything
 - Units of ~30mins worth of data
- PASS1 Reconstruction run on spooled data
 - Pseudo real-time
 - Several Reconstruction nodes (~10)
 - Added reconstruction data expected to be 2x raw data (50KB)
- Baseline model is to use flat files for spooling
- Should we be writing into OODBMS directly from Level 3?
 - Advantages:
 - ~20% decrease in total bandwidth through PASS1 (avoids extra output of raw data following reconstruction)
 - Uniformity - OODBMS is only storage medium
 - Disadvantage of “real” real-time rather than pseudo real-time
 - Limited upstream buffering to smooth over problems

David Quarrie - Proposal for a OODBMS-based Event Store

Event Collections

- 10^9 objects not manageable in a single collection
 - Not a space issue
 - Even if only an event header is kept for each event
- Need Hierarchy of Collections (collections of collections)
 - Ideally have transparent iterators, but not essential
- Natural Granularity
 - Runs
 - Periods of order of hours of “stable” conditions
 - Management unit - a convenience in referring to data dating interval
 - Run Periods
 - Periods of several weeks/months of “stable” accelerator/detector conditions
 - Laboratory scheduling unit
- Use natural granularity?
 - Look at implications on space

David Quarrie - Proposal for a OODBMS-based Event Store

Run-based collections

- Raw Data: 25KB per event at 100Hz
 - Second: 100ev; 2.5MB
 - Hour: 3.6×10^5 ev; 8GB
 - Day: 9×10^6 ev; 200GB
- Multiple data writers (~10)
 - Level 3 nodes
 - PASS1 Reconstruction Nodes
- Multiple data readers (~10)
 - PASS1 Reconstruction Nodes
- Natural parallelism
 - Decreases database size per writer
 - ~800MB/hour raw data
 - ~1.6GB/hour for reconstructed data
 - Decreases bandwidth per writer/reader
 - 250KB/sec for raw data
 - 500KB/sec for reconstructed data

David Quarrie - Proposal for a OODBMS-based Event Store

Event Collections

- Choose a database size
 - Of order of 100's of MB to 10's of GB
 - Following discussion based on 1GB
 - ~80 mins of raw data per writer
 - ~40 mins of reconstructed data per writer
 - Only allows ~30TB total (32k databases per federated database limit)
 - Probably a bit small but reasonable for discussion purposes
- Hierarchy
 - Experiment (10's of Run Periods)
 - Timescale of years
 - Run Period (100's of Runs)
 - Timescale of months
 - Run (10's of Databases)
 - Timescale of hours
 - Databases (~ 5×10^5 events)
 - Timescale of 10's of minutes
- All components in hierarchy necessary? (e.g. Run Period)

David Quarrie - Proposal for a OODBMS-based Event Store

Event Collection Summary

- Write ~1GB databases directly from each Level 3 node and PASS1 reconstruction node
 - Allows spreading bandwidth across multiple disk spindles
- Subdivide into 100MB Containers within database
 - Locking granularity
 - 4 minutes latency before reconstruction could start on raw data
 - MROW would allow even less latency
- Write fixed size databases
 - Optimises mass store utilisation
 - Long runs span multiple databases
 - Multiple short runs per database
 - Requires a catalog of mapping of runs to databases
- Event collections use natural hierarchy
 - Experiment, Run periods, runs
- Need hierarchical iterators

David Quarrie - Proposal for a OODBMS-based Event Store

Event Structure

- Goals
 - Extensible event structure
 - Reconstruction is essentially process of decorating an event
 - Support different clustering strategies
 - Avoid schema evolution
 - It will be necessary but only if we change our minds or forget something
- Many items within an event are collections
 - List of tracks
 - List of vertices
 - List of drift chamber hits
 - etc.
- Treat Event as a collection of collections
 - Make queries on event to locate collections

David Quarrie - Proposal for a OODBMS-based Event Store

Event Structure

- Take advantage of phases
 - Raw data
 - Reconstructed data
 - DST data
 - etc.
- Design top-level Event object as containing a VArray, with each element being an *ooRef* to another object corresponding to a processing phase.
 - Naturally extensible to accommodate more phases
 - Use of *ooRef* allows all other data to be elsewhere (other databases)
- Easy to locate data for a particular phase
 - Fixed offset, linear search, small hash table
 - Choose something
- Top-level Event object is small
 - Could contain some summary information as well as VArray

David Quarrie - Proposal for a OODBMS-based Event Store

Event Structure (Cont.)

- Design Processing Phase objects as containing VArrays, with each element being an *ooRef* to another object containing the actual collections
 - Naturally extensible to accommodate more collections
 - e.g. Raw Data object
 - Each VArray element *ooRef*'s an object containing to data from a detector subsystem
 - e.g. Reconstructed Data object
 - Each VArray element *ooRef*'s an object containing the list of tracks, or the list of vertices or....
 - Processing Phase objects can themselves contain summary information
- Easy to locate a particular component
 - Fixed offset, linear search, small hash table
 - Choose something
- Use of *ooRef*'s allows different clustering strategies

David Quarrie - Proposal for a OODBMS-based Event Store

Summary Information

- Can be kept at several levels
 - Event Header
 - Processing Phase Objects
 - Track List object
 - Detector Subsystem Object
- Useful for making queries with minimal data access
 - e.g. query based on number of tracks doesn't have to loop over tracks in track list to determine the number
- Most useful when supported by predicate language
 - Access to both data and function members
- Model is to select events on summary information, avoiding accessing detailed information
 - Hierarchy of summary information allows trade off between performance and complexity

David Quarrie - Proposal for a OODBMS-based Event Store

Clustering Strategies

- Event headers in hierarchical collections as discussed earlier
 - Small
 - Lend themselves to replication for particular physics signatures
 - Note: Iterators for these collections need to know where they are in the hierarchy so that the application just "sees" a sequence of events.
- Raw Data for each event clustered together
 - Each Level 3 node has information for complete event
 - Normal mode of access is by PASS1 and most of the information per event is accessed
 - Optimal for fast database population
- Data clustering for other phases?
 - Either cluster by event or by type
 - Needs better understanding of access patterns
 - Generally migrate from by-event to by-type for later phases
 - Might need to replicate data in both strategies

David Quarrie - Proposal for a OODBMS-based Event Store

Collections of physics events - headers

- Event Header collection in sequence of databases
 - Space per event ~40bytes + summary information (say 200 bytes)
 - 10^7 event sample occupies ~2.5GB
- Raw Header collection in sequence of databases
 - Space per event ~200bytes + summary information (say 1kb)
 - 10^7 event sample occupies ~12.5GB
- Reconstructed Header collection in sequence of databases
 - Space per event ~200bytes + summary information (say 1kb)
 - 10^7 event sample occupies ~12.5GB
- DST Header collection in sequence of databases
 - Space per event ~200bytes + summary information (say 1kb)
 - 10^7 event sample occupies ~12.5GB
- etc.

David Quarrie - Proposal for a OODBMS-based Event Store

Collections of physics events - data

- Raw Data collection in sequence of databases
 - Space per event ~25kb
 - 10^7 event sample occupies 250GB
- Reconstructed Data collection in sequence of databases
 - Space per event ~50kb
 - 10^7 event sample occupies ~500GB
- DST Data collection in sequence of databases
 - Space per event ~5kb
 - 10^7 event sample occupies ~50GB
- MicroDST Data collection in sequence of databases
 - Space per event ~1kb
 - 10^7 event sample occupies ~10GB

David Quarrie - Proposal for a OODBMS-based Event Store

Migration, Replication etc.

- Need to have concentrated data
 - Access most events in event sample
 - Access most data in event
 - Clustering can only improve performance in one dimension
- Expect to replicate data
 - Event samples with particular physics signatures
 - Need to reconcentrate data when it's dilute
 - Based on access patterns
- Replication strategy for remote access

David Quarrie - Proposal for a OODBMS-based Event Store

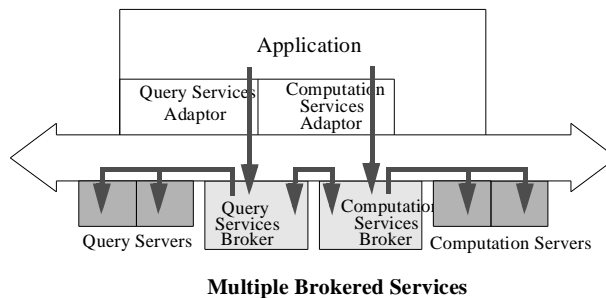
Mapping from existing Event API

- AbsEvent API designed to accommodate this
- HepAList to VArray?
- No pointers?
 - Replace by references?
 - Chris Day has a prototype `d_Ref<T>` that behaves correctly
- Basic data types
 - int, float, etc. supported
 - int16, int32, float32, etc. currently preferred by Objectivity (3.8)
 - `d_Long`, `d_Float` etc. is ODMG-93 standard (Objy 4.0?)
- How to create persistent objects
 - Cannot use “new” operator
 - Need overloaded “new(...)” with clustering hints etc.
 - Factory objects?
 - Subclass all persistent objects from transient classes & use copy semantics?

David Quarrie - Proposal for a OODBMS-based Event Store

Query Servers

- Definition
 - Given an input collection (of collections) and a predicate, return an output collection (of same type) of elements that satisfy the query
 - Essentially ODMG-93 query
- Wish to parallelise the query
- Plan is to use PASS Architectural Model (CORBA)
 - OID to output collection (of collections) in OODBMS is result



David Quarrie - Proposal for a OODBMS-based Event Store

Regional and Institution Centers

- Objectivity/DB supports concept of Partitions
 - Replication of parts of Federated Database
 - Database catalog, schema catalog
 - Lock Server
 - Databases
- Database replication by network or tape copy
 - Install a backup copy of set of databases on remote site
 - Objectivity/DB will automatically update changes since backup made
- Remote sites use their replicated databases if available
 - Access non-replicated databases remotely otherwise
- Updates to databases propagated everywhere
 - Normally instantaneous but at speed of slowest
 - Can be deferred by declaring sites "offline"

David Quarrie - Proposal for a OODBMS-based Event Store

Regional and Institution Centers

- Partition per center (~80 within BaBar)
- Decide on which databases to distribute
 - All Header databases & DST data etc. to Regional centers?
 - Only Event Header & microDST data to Institution?
- Use network or tape backup to install database replicas
 - Let Objectivity/DB upgrade them to latest state
- Decide on update frequency
- Declare all sites offline until update required
- Simple model is single tier (Institution falls back to Collaboration center for remote data)
 - Needs more R&D to see whether 2-tier possible
- Performance issues?

David Quarrie - Proposal for a OODBMS-based Event Store

Plans (tentative)

- Implement the BaBar Event Structure based on this extensible model
 - Details hidden from application programmers via abstract API
 - Requires some concrete implementation classes
 - Timescale April-May 1996
- Implement an Input Module within BaBar framework to access OODBMS
 - (c.f. CDF Analysis_Control)
 - Application Modules independent of source of data
 - “Next event” just locates the next Event Header object in the collection
- Populate a small (GB's) database to test Event Collection hierarchy strategy
 - Timescale June-July 1996

David Quarrie - Proposal for a OODBMS-based Event Store

Plans (Cont.)

- Investigate use of SLAC/NERSC for large-scale prototypes
 - NERSC & PDSF located at LBNL
- BaBar decision on adoption of this technology
 - Nov 1996
- Understand integration with hierarchical mass store
 - Early 1997?
- Prototype BaBar implementation available at SLAC
 - Nov 1997
- Production BaBar implementation available at SLAC
 - Dec 1998

David Quarrie - Proposal for a OODBMS-based Event Store

Status & Issues

- Issues
 - Manpower
 - ~ 1FTE available in 1996 (not enough)
 - Licensing
 - Restricted to SLAC?
 - Regional centers?
 - Run-time vs application developers licenses
 - Objectivity support for ODMG-9x
 - Objy predicate language only allows access to data members
 - Support for ODMG queries
 - Iterators on VArrays
 - etc.
- Status
 - Programmer evaluating Objectivity for calibration constants database
 - Hope to use at least 50% on data storage project
 - Some PASS close-out funds still available
 - Need to discuss use in this context (~ 6-9 programmer-months)
 - Some of my time & Chris Day's time (both worked on PASS)
 - Preliminary design underway
 - This talk pretty much summarises it

David Quarrie - Proposal for a OODBMS-based Event Store