O'REILLY®
EUROPEAN
OPEN
SOURCE
CONVENTION
17-20 OCTOBER 2005

Building Berkeley DB XML from Open Source Components

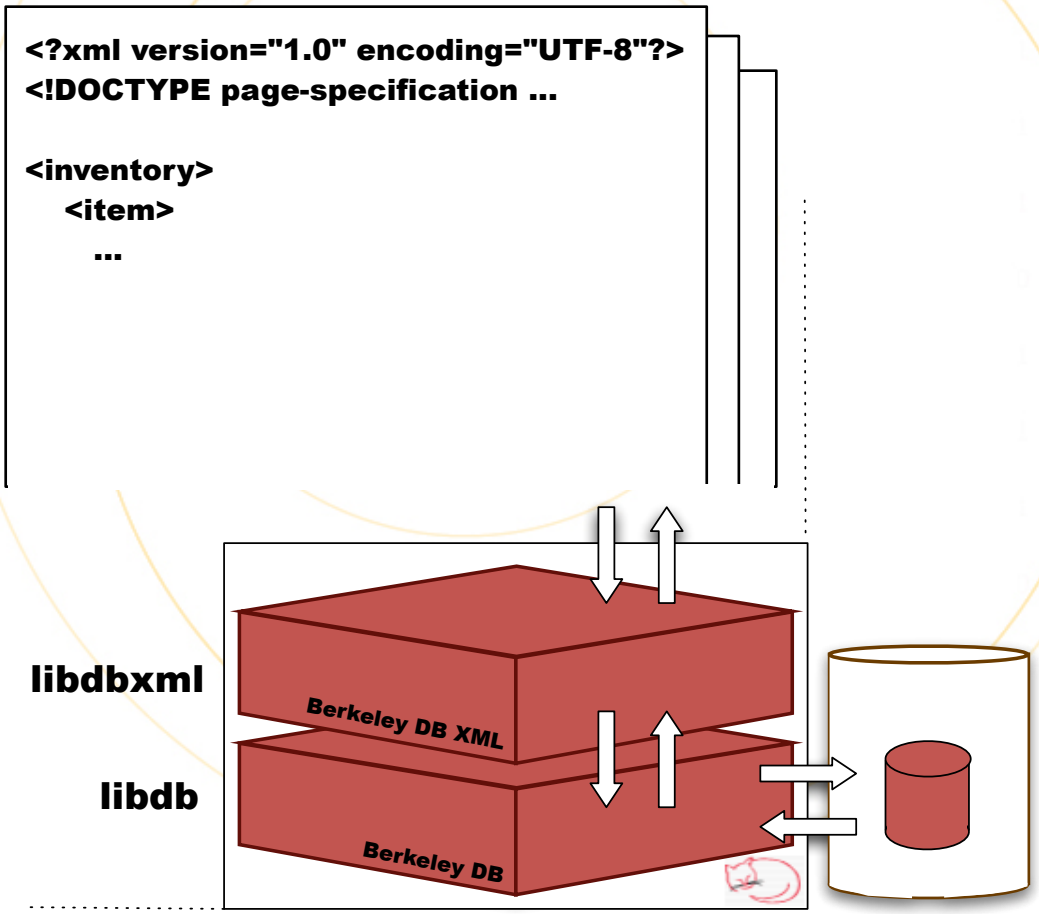
Gregory Burd • Product Manager • gburd@sleepycat.com

Agenda

1. Learn from our mistakes, our pain need not be your pain.

2. This is not a code tutorial, nor is it a business yadda yadda or a sales pitch.

The Concept



The Components

Technical Elements

- store
  - efficient, compact, maintainable
- parse
  - examine, validate
- index
  - flexible, mutable, extensible
- query
  - process, plan, optimize, execute

Technical Elements

- What we had:
  - store
    - efficient, compact, maintainable
- What we lacked:
  - parse
    - examine, validate
  - index
    - flexible, mutable, extensible
  - query
    - process, plan, optimize, execute

Berkeley DB XML 0.8

- store Berkeley DB 4.1
- parse Expat 1.2
- index in house
- query in house

store: **Berkeley DB**

- Key/data transactional storage
- Our IP, no issues there
- Need a data representation

issue: Missing a concurrent sequence generator for document IDs.

parse: **Expat**

- Stream based XML parser
- MIT License
- Written in C

issue: Stream based representation of XML documents is of no value when evaluating a query.

index: **In House**

- This will become the core asset
- No existing standard like SQL
- Must be flexible, extensible

issue: Where to begin? The options were boundless, we had to narrow things down to a manageable set.

query: **In House**

- XPath
- This was a prototype, it only partially worked.

issue: This could be a unknown amount of development time, it must be carefully managed and bounded somehow.

Berkeley DB XML 0.8

- Never released to the public
- A Native XML Database (NXD) library
- XPath 1.0
- Mature, Transactional Storage
  - Transactions, Encryption, XA, etc.
- C++ API
- And it worked!

Lessons from the Prototype

- We needed a way to manage the representation of XML during query processing.
- We need DB to add some features, fix some bugs, and add some API.
- Our value is in fast, transactional access to XML.

Berkeley DB XML 1.1

- store Berkeley DB 4.2
- parse Xerces-C
- index in house
- query Pathan

store: **Berkeley DB**

- How will the data be stored?
- What is the norm for XML data?
- Most efficient representation?

issue: Do we expect lots of small documents, or one large document? Do we store the entire document in a single key/data pair or break it up into the document's nodes?

parse: **Xerces**

- Arrived with Pathan, made Expat redundant
- Stream or DOM parsing

issue: If we are going to swap out Expat, let's make sure we pick the right thing. What about libxml2? Is C++ the right call?

index: **Our Code**

- This is to become our central asset, let's do a good job.
- Queries will examine the DOM, indexes built from SAX analysis.

issue: What is a good API and conceptual indexing model that can support a changing implementation?

query: **Pathan**

- Parse and evaluate XPath expressions
- Requires the Xerces-C DOM interface

issue: Does this project have a open source community that will actively own and develop in partnership with us over time?

Berkeley DB XML 1.1.x

- A Native XML Database (NXD) library
- XPath 1.0
- Mature, Transactional Storage
  - Transactions, Replication, Encryption, XA, etc.
- Millions, Terabytes, Sub-second
- Minimal or no administration
- C++, Java, TCL, & Python APIs
- First commercial version

Lessons from 1.1.x

- They wanted XQuery, we had XPath
- Documents were larger than anticipated
- We needed a “first customer”

Berkeley DB XML 2.x

- store Berkeley DB 4.3
- parse Pathan 2.0 (XPath 2.0)
- index in house
- query Progress Software XQuery

store: **Berkeley DB**

- We need two storage models, node and whole document.
- Work leads to a 30x speed up.

issue: So much changed that we knew we'd need to make many small releases to get things tuned properly. Would a rapid release cycle turn users off?

parse: **Pathan**

- We found it a bit lonely
- Lots of fixes to the code
- Memory and error code issues

issue: How do we keep from forking or if we fork how do we move the work back into the community when the community is starting to fade?

index: **Our Code**

- This will be one of our central assets, let's do a good job.
- Index XML during the parse phase using SAX.

issue: With XQuery, which includes XPath, does our indexing design still cover all the necessary cases?

query: **XQuery**

- Pathan + XQuery
- We're going to fork
- Is this core or can we donate it?

issue: What will happen with the new Pathan/XQuery source that results from our merger? We've forked a languishing project.

Berkeley DB XML 2.x

- A Native XML Database (NXD) library
- XQuery 1.0 July 2004 Draft
- Mature, Transactional Storage
  - Transactions, Replication, Encryption, XA, etc.
- Millions, Terabytes, Sub-second
- Minimal or no administration
- Command line shell
- C++, Java, TCL, PHP & Python APIs
- Introductory Guided Tour Document
- Manages any document size well
- Much easier installation

Lessons from 2.x

- XQuery is the right choice
- Node level storage works
- Ease of use is critical
  - command line
  - introductory guides
  - easy installation
    - especially for Windows users
  - Java, Perl, Python, and PHP
  - .NET/C#, XML:DB and servers

The XML Storage Market

- Telecom/Internet Services
- Configuration
- Integration
- SOA
- Biotech
- Homeland Security
- Content Management

Starwood Hotels

Searching for a nice place to stay?

If you're searching Starwood, you're using Berkeley DB XML.

Juniper

The Juniper NetScreen Security Manager has to determine what is a threat,

the threat reports live in Berkeley DB XML.

Zeliade

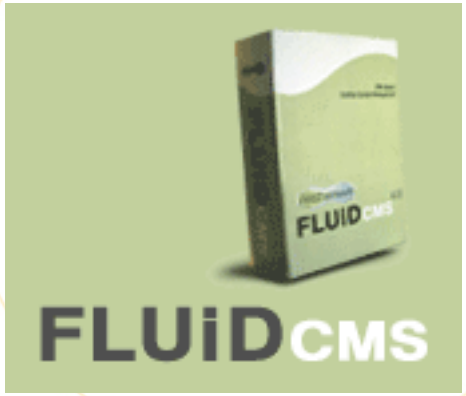
Mass amounts of data for financial models models in XiMoL

XML objects reside in Berkeley DB XML.

zeliade systems

Feedstream

Zero programming next generation XML based content management...



and the content resides in Berkeley DB XML.

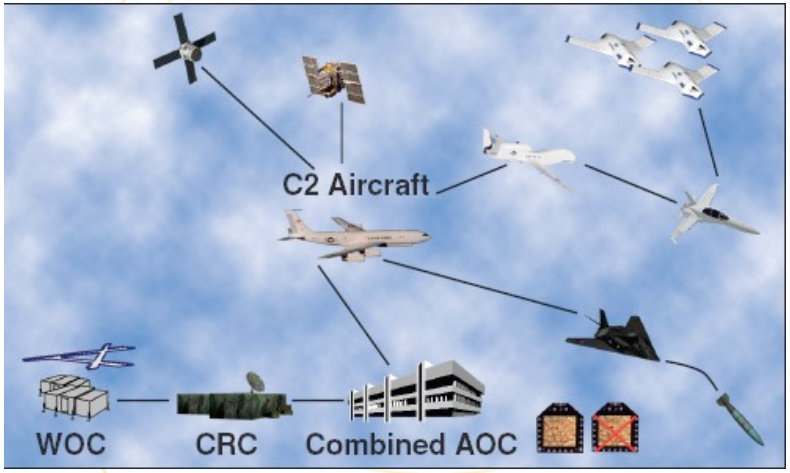
Berkeley Medical

Your prescription drug history may be stored as XML...

in Berkeley DB XML.

USAF Research Labs

Publish and subscribe integration with XML messages,



where the data moves through Berkeley DB XML.

AFRL
THE AIR FORCE RESEARCH LABORATORY
LEAD | DISCOVER | DEVELOP | DELIVER

Syncato

An open source XML content publication system written in Python,

built on top of Berkeley DB XML.

Syncato

A content management system for the little things

Developing Markets

- Higher education
- Research
- Notification
- The ‘X’ in LA~~M~~P

O'REILLY®
EUROPEAN
OPEN
SOURCE
CONVENTION
17-20 OCTOBER 2005

Go Native!

Choose the right database for your XML data.

Gregory Burd • Product Manager • gburd@sleepycat.com

Database?

YAHOO! RESEARCH

BUZZ GAME

http://buzz.research.yahoo.com/