

UFSQL: Unified File System Query Language

Yan Li , Yunfei Chen

<yanli@ucsc.edu>, <ychen@soe.ucsc.edu>

Mar 8, 2012

Jack Baskin School of Engineering
University of California, Santa Cruz

Why file system query is
important?

Query! Don't Browse!

- ✦ Exploding amount of information
- ✦ Complex and plethora metadata
- ✦ Fading human memory
- ✦ Finding the right information is hard

Anatomizing FS Query

3 Classes of File System Query

- ✦ Class 1: Free text query
- ✦ Class 2: Structured metadata query
- ✦ Class 3: Provenance query

Class 1: Free Text

- ✦ Files with “California” in contents
- ✦ Files with “tax” in file name
- ✦ Files with “usr” in path name
- ✦ Files with “rms” in uid

Class 2: Structured Metadata

Modern file systems can have complex hierarchical metadata



EXIF

ExifVersion, Manufacturer, Model, Orientation, Date and Time, Height, Width, ...

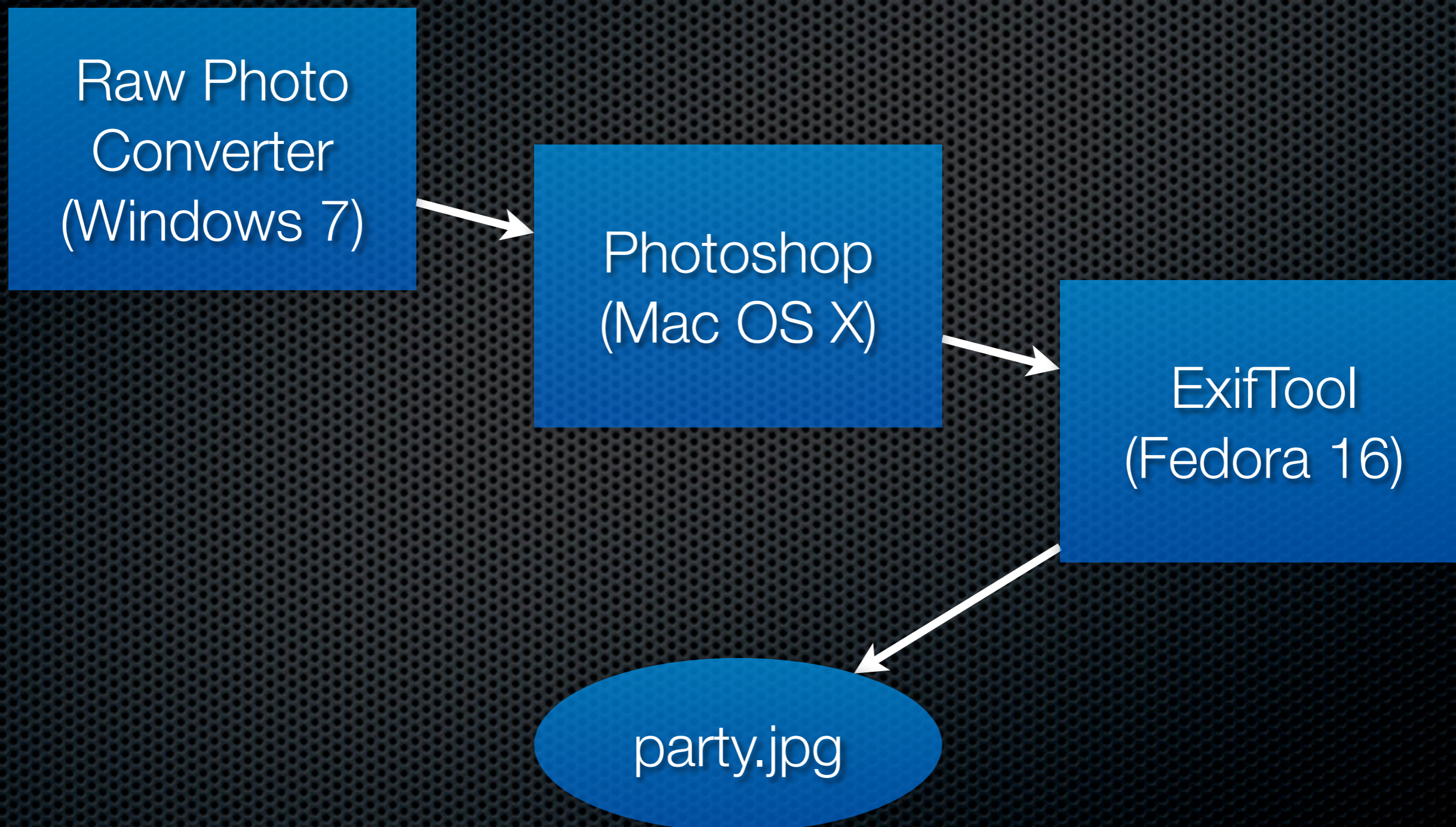
MakerNote

MacroMode, Quality, Metering Mode, ...

Geolocation

Longitude, Latitude, Altitude, Hemisphere, ...

Class 3: Provenance Graph



Why we need a new query language?



Today's Database
and Structured Data
Query

Today's File System
Query

Designing a new FS Query Language

Unified File System Query Language: UFSQL

- ✦ Using Functional Programming model (like XQuery)
- ✦ Programmable
- ✦ Extensible
- ✦ Supports 3 classes of queries

A Complex Sample

We are running a photo studio.

We have a photo gallery.

How many pictures each photographer took and processed in the last month?

How to Get File Author?

Different camera / software produces different metadata:

- ✦ Canon: `MakerNote:Author`
- ✦ Nikon: `EXIF:ImageComment`
- ✦ Photoshop: `XMP:Author`

Solution 1

```
AuthorField =  
  for PhotoStorage.metadata  
  where  
    field.name in ["MakerNote:Author",  
                  "EXIF:ImageComment",  
                  "XMP:Author"]  
  return field
```

Solution 2

```
GetAuthor file =  
  for file.metadata join fieldNames  
  let  
    fieldNames = ["MakerNote:Author",  
                  "EXIF:ImageComment",  
                  "XMP:Author"]  
    authorList = file.metadata  
  where authorList.value <> NIL  
  return authorList
```

How many pictures each photographer took and processed in the last month?

```
AuthorField = (...)
```

```
for PhotoStorage.* as file
```

```
where
```

```
    Now() - file.modifiedTime <
```

```
        DateRange(0,1,0) and
```

```
    file.prov.* contains "photoshop"
```

```
group by GetAuthor(file)
```

```
return GetAuthor(file) as Author, Count(*) as
```

```
    PhotoCount
```


How many pictures each photographer took and processed in the last month?

AuthorField = (...) ← **Functional Programming Model**

for **PhotoStorage.*** as **file**

where

Now() - **file.modifiedTime** <

DateRange(0,1,0) and

file.prov.* contains “photoshop”

group by GetAuthor(**file**)

return GetAuthor(**file**) as **Author**, Count(*) as

PhotoCount

How many pictures each photographer took and processed in the last month?

AuthorField = (...) ← **Functional Programming Model**

for **PhotoStorage.*** as **file**

where

Now() - **file.modifiedTime** <

DateRange(0,1,0) and

file.prov.* **contains** "photoshop"

group by **GetAuthor(file)**

return **GetAuthor(file)** as **Author**, **Count(*)** as

PhotoCount

Dynamic Graph Scanning

How many pictures each photographer took and processed in the last month?

AuthorField = (...) ← Functional Programming Model

for **PhotoStorage.*** as file
where

Now() - **file.modifiedTime** <

DateRange(0,1,0) and

file.prov.* **contains** "photoshop"

group by GetAuthor(**file**)

return GetAuthor(**file**) as **Author**, Count(*) as
PhotoCount

Dynamic Graph Scanning

↑ Returns complex data

Basic Syntax of UFSQL

Basic Syntax: FLOWR

QUERY FUNCTION =

for <FILE_SET>

let <VAR_BIND>

where <EXPRESSIONS>

order by <EXPRESSIONS>

return <VARS>

Modules

```
import Graph.Face
```

```
for PhotoStorage.* as file
```

```
where
```

```
  faceList(file) contains "Joe"
```

```
group by faceList(file)
```

```
return file.name
```

Our Innovation

- ✦ First **functional query language** for file systems
- ✦ First query language that combines **3 classes of queries**: free text, structured data and provenance graph query
- ✦ Extensible by using **modules**
- ✦ In-function SQL-like dataset **join**

Future Research

- ✦ More strong provenance query support (need more feedback from the end-user)
- ✦ A test prototype engine
- ✦ A natural language layer for ease of use

Related Research

Old FS Query Language

- ✦ Primitive tools: ls, find (1980s)
- ✦ Spotlight / Windows Search (2005)
- ✦ XQuery (2007)
- ✦ QUASAR: replacing file system interfaces with a query language (2008)
- ✦ Harvard PQL: Path Query Language (2008)

Thank you!

Acknowledgment

We would like to thank the faculty and students of the Storage Systems Research Center (SSRC), Center for Research in Intelligent Storage (CRIS), and IRKM lab for their help and guidance. This research was supported in part by the National Science Foundation, and by the industrial sponsors of the SSRC and CRIS, including EMC, Hewlett Packard Laboratories, IBM Research, NetApp, Northrop Grumman, and Samsung.

Copyright notice

- The red sports car is from Peugeot, the tractor is from BC
- Trademark Acknowledgment:
All other products or services mentioned in this document are identified by the trademarks, service marks, or product names as designated by the companies who market these products. Inquiries concerning such trademarks should be made directly to those companies.