

LA-UR-12-23717

Approved for public release; distribution is unlimited.

Title:	Applied Parallel Metadata Indexing
Author(s):	Jacobi, Michael R
Intended for:	4th Annual Computing and Information Technology Student Mini-Showcase, 2012-08-02 (Los Alamos, New Mexico, United States)



Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Michael Jacobi
University of New Mexico
HPC-3
Cody Scott/Aaron Torres

Applied Parallel Metadata Indexing

Abstract: The GPFS Archive is a parallel archive used by hundreds of users in the Turquoise collaboration network. It houses 4+ petabytes of data in more than 170 million files. Currently, users must navigate the file system to retrieve their data, requiring them to remember file paths and names. A better solution might allow users to tag data with meaningful labels and search the archive using standard and user-defined metadata, while maintaining security. Last summer, I developed the backend to a tool that adheres to these design goals. The backend works by importing GPFS metadata into a MongoDB cluster, which is then indexed on each attribute. This summer, I implemented security and developed the user interface for the search tool. To meet security requirements, each database table is associated with a single user, which only stores records that the user may read, and requires a set of credentials to access. The interface to the search tool is implemented using FUSE (Filesystem in Userspace). FUSE is an intermediate layer that intercepts file system calls and allows the developer to redefine how those calls behave. In the case of this tool, FUSE interfaces with MongoDB to issue queries and populate output. A FUSE implementation is desirable because it allows users to interact with the search tool using commands they are already familiar with. These security and interface additions are essential for a usable product.

Applied Parallel Metadata Indexing

Mike Jacobi
HPC-3

Mentors:
Aaron Torres
Cody Scott

Use Case

- ⊗ Users generate data
- ⊗ Locating this data is difficult
- ⊗ Search for it, instead of remember it
- ⊗ Extended attributes – tag your records

GPFS Archive



GPFS Policy

```
RULE EXTERNAL LIST 'listtest' EXEC  
'list_exec.py'  
RULE 'r2' LIST 'listtest' DIRECTORIES_PLUS  
SHOW( VARCHAR(XATTR(tag)) || ',' || VARCHAR  
(MODE) || ',' || VARCHAR(USER_ID) || ',' || VARCHAR  
(GROUP_ID) || ',' || VARCHAR  
(FILE_SIZE) || ',' || VARCHAR(DAYS  
(MODIFICATION_TIME)) || ',' || VARCHAR(DAYS  
(CREATION_TIME)) || ',' || VARCHAR  
(BLOCKSIZE))
```

Policy
Processing
Script

Bulk Insert
into MongoDB

Databases of
Metadata

MongoDB
Interface

Query
"find x<y"

FUSE

Server 1

Shard 1

Server 2

Shard 2

Server N

Shard N

Query Output

```
/archive/old/s/script.py  
/archive/old/s/reet.csv  
/archive/old/s/todo.txt  
/archive/old/s/support.c ...
```

Creating Security

- ⊗ Databases disregard file system security
 - ⊗ But we must retain it
- ⊗ Database per user
 - ⊗ Authentication into databases
- ⊗ A record is inserted into a db only if user has read access
 - ⊗ How do you know if you can read a file?
- ⊗ Policy processing script receives records (in parallel)
 - ⊗ Sort by path in alphabetical order
 - ⊗ Maintain a list of sets of users with read access
 - ⊗ Determine a record's read accessible users

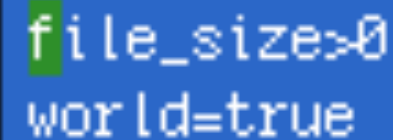
Enforcing Security

- ⊗ Why use FUSE? File systems are good at security
 - ⊗ My custom solution would not be
- ⊗ How does it work? Intercepts file system calls and lets the developer redefine how those functions behave
- ⊗ Provides an intuitive user interface
 - ⊗ Smart folders!
- ⊗ I used FUSE to interface with Mongo - MongoFS

Create a Query

```
[root@gpfs03 mongo]# ls -al fusemount/  
total 8  
drwxr-xr-x 3 root root 2048 Jul 31 10:15 .  
drwxr-xr-x 6 root root 2048 Jul 31 10:13 ..  
drwxr-xr-x 2 root root 2048 Jul 30 19:20 a  
-rw-r--r-- 1 root root    11 Jul 27 16:51 .query  
[root@gpfs03 mongo]# mkdir fusemount/mytestquery  
[root@gpfs03 mongo]# ls -al fusemount/mytestquery/  
total 4  
drwxr-xr-x 2 root root 2048 Jul 31 10:16 .  
drwxr-xr-x 4 root root 2048 Jul 31 10:16 ..  
-rw-r--r-- 1 root root    0 Jul 31 10:16 .query  
[root@gpfs03 mongo]# vi fusemount/mytestquery/.query
```

Query File



```
file_size>0  
world=true
```

Issue the Query

```
[root@gpfs03 mongo]# ls fusemount/mytestquery/  
5.8.7.5018076b028b172017c46b5f  File.5018076b028b172017c46bd6  
5.8.7.5018076b028b172017c46b9d  File.5018076b028b172017c46bf3  
a.5018076b028b172017c46b9f      Filter.5018076b028b172017c46b43  
ANSIColor.5018076b028b172017c46b95  Filter.5018076b028b172017c46b70  
ANSIColor.5018076b028b172017c46beb  Filter.5018076b028b172017c46b85  
archive.W.5018076cbdc3dbb9782de826  Filter.5018076b028b172017c46bb4  
arpa.5018076b028b172017c46b40      Filter.5018076b028b172017c46bd8  
Attribute.5018076b028b172017c46b60  g.5018076b028b172017c46bf9  
Attribute.5018076b028b172017c46ba4  Getopt.5018076b028b172017c46b72
```

```
[root@gpfs03 mongo]# ls fusemount/mytestquery/ | wc -l  
199
```

.query File

```
file_size<100  
world=true  
~  
~
```

```
file_size<5  
file_size>=3  
world=true
```

Output

```
[root@gpfs03 mongo]# ls -al fusemount/mytestquery/.query  
total 2  
-rw-r--r-- 0 root root 2 Jul 31 2012 a.5018076b028b172017c46b9f  
-rw-r--r-- 0 root root 3 Jul 31 2012 b.5018076b028b172017c46ba0  
-rw-r--r-- 0 root root 4 Jul 31 2012 c.5018076b028b172017c46ba1  
-rw-r--r-- 0 root root 5 Jul 31 2012 d.5018076b028b172017c46bf6  
-rw-r--r-- 0 root root 6 Jul 31 2012 e.5018076b028b172017c46bf7  
-rw-r--r-- 0 root root 7 Jul 31 2012 f.5018076b028b172017c46bf8  
-rw-r--r-- 0 root root 8 Jul 31 2012 g.5018076b028b172017c46bf9  
-rw-r--r-- 1 root root 25 Jul 31 10:43 .query  
[root@gpfs03 mongo]#
```

```
[root@gpfs03 mongo]# ls -al fusemount/mytestquery/.  
total 0  
-rw-r--r-- 0 root root 3 Jul 31 2012 b.5018076b028b172017c46ba0  
-rw-r--r-- 0 root root 4 Jul 31 2012 c.5018076b028b172017c46ba1  
-rw-r--r-- 1 root root 36 Jul 31 10:53 .query  
[root@gpfs03 mongo]#
```

World Readable

```
file_size>0  
world=true
```

```
d-wxr-xr-x 0 root root 8192 May 14 21:33 Filter.5018076b028b172017c46bb4  
d-wxr-xr-x 0 root root 8192 May 14 21:33 Filter.5018076b028b172017c46bd8  
-rw-r--r-- 0 root root 8 Jul 31 2012 g.5018076b028b172017c46bf9  
d-wxr-xr-x 0 root root 8192 May 14 15:33 Getopt.5018076b028b172017c46b72  
d-wxr-xr-x 0 root root 8192 May 14 21:33 Getopt.5018076b028b172017c46bda  
d-wxr-xr-x 0 root root 8192 May 14 21:33 Glob.5018076b028b172017c46bb3  
drwxr----x 0 root root 8192 Jul 31 2012 gpfstant.V.5018076cbdc3d8b9782de825  
d-wxr-xr-x 0 root root 8192 May 14 15:33 Handlers.5018076b028b172017c46b61  
d-wxr-xr-x 0 root root 8192 May 14 21:33 Handlers.5018076b028b172017c46ba5  
d-wxr-xr-x 0 root root 8192 May 14 21:33 Harness.5018076b028b172017c46bed  
d-wxr-xr-x 0 root root 8192 May 14 21:33 HiRes.5018076b028b172017c46bc8  
d-wxr-xr-x 0 root root 8192 May 14 21:33 Hostname.5018076b028b172017c46bc4
```

Multiple Query Clauses

```
[root@gpfs03 mongo]# cat fusemount/mytestquery/.query  
creation_time>1338045200  
file_size<100  
blocksize>0  
user_id=0  
group_id=0  
name=g  
world=true  
[root@gpfs03 mongo]# ls -al fusemount/mytestquery/  
total 0  
-rw-r--r-- 0 root root 8 Jul 31 2012 g.5018076b028b172017c46bf9  
-rw-r--r-- 1 root root 90 Jul 31 11:36 .query
```

Reading and Copying

```
[root@gpfs03 mongo]# cat fusemount/mytestquery/c.5018076b028b172017c46ba1
abc
[root@gpfs03 mongo]# cp fusemount/mytestquery/c.5018076b028b172017c46ba1 /tmp/dummy
[root@gpfs03 mongo]# cat /tmp/dummy
abc
[root@gpfs03 mongo]#
```

Change Mode

```
[root@gpfs03 mongo]# ls -l fusemount/
total 4
drwxr-xr-x 2 root root 2048 Jul 30 19:20 a
drwxr-xr-x 2 root root 2048 Jul 31 10:53 mytestquery
[root@gpfs03 mongo]# chmod 700 fusemount/mytestquery/
[root@gpfs03 mongo]# ls -l fusemount/
total 4
drwxr-xr-x 2 root root 2048 Jul 30 19:20 a
drwx----- 2 root root 2048 Jul 31 10:53 mytestquery
[root@gpfs03 mongo]# chmod 700 fusemount/mytestquery/c.5018076b028b172017c46ba1
chmod: changing permissions of 'fusemount/mytestquery/c.5018076b028b172017c46ba1': Operation not permitted
[root@gpfs03 mongo]#
```

Xattrs

```
[root@gpfs03 mongo]# attr -l fusemount/mytestquery/c.5018076b028b172017c46ba1  
Attribute "path" has a 32 byte value for fusemount/mytestquery/c.5018076b028b172017c46ba1
```

```
[root@gpfs03 mongo]# attr -g path fusemount/mytestquery/c.5018076b028b172017c46ba1  
Attribute "path" had a 32 byte value for fusemount/mytestquery/c.5018076b028b172017c46ba1:  
/gpfsmnt/archive/mjacobi/small/c
```

Security

```
[root@gpfs03 mongo]#  
[root@gpfs03 mongo]# ls -al fusemount/mytestquery/ | wc -l  
201  
[root@gpfs03 mongo]# su mjacobi  
[mjacobi@gpfs03 mongo]$ ls -al fusemount/mytestquery/ | wc -l  
11  
[mjacobi@gpfs03 mongo]$
```

Future Work

- ⊗ Handling directories within FUSE
- ⊗ Allowing write access
- ⊗ Naming conventions – get rid of Mongo id
- ⊗ Xattr indexing and querying
- ⊗ Insert optimization – more hardware?

Other Fun Things

- ⊗ Software Insert optimization
- ⊗ IBM bug report
- ⊗ The Search Tool config file
- ⊗ Python to C adventures