ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 5; Issue 3; May-June: 2017; Page No. 96-101
Journal Approved by UGC

Meta Data File Management System using BigData

Dr.Ch .GVN Prasad[1], M. yashoda[2], A.Mamatha[3]

[1]Professor and HOD in Depart of CSE. Sri Indu College of engineering and Technology

[2]PG Scholar in Department of CSE. CSE. Sri Indu College of engineering and Technology

[3]PG Scholar in Department of CSE. CSE. Sri Indu College of engineering and Technology

Abstract

Implement a modular metadata system on existing file system stack for quick search, storage and retrieval of data present in different types of files. Current Big data Operating Systems do not support such functionality. Only file specific applications like Picasa, iTunes, and iPhoto support searching specific file types. From a user perspective, it becomes very important as the number of photos, songs and documents we store on our disks increase. A disk search also is very

slow. The current proposal willallow large number of data to be searched and retrieved based on metadata in a very short time. The current operating systems do support the current feature (like OS X) with the support of tags, but the functionality is not available at a file system level. We will add programmatic API support to add metadata information on the file system such that extensions can be written around these APIs to extend the addition/deletion of attributes to a file

system entity. The access times will be compared with existing filesystem's metadata access time.

Keywords: Traditional, Spreadsheet, Metadata, bigdata

Introduction

Metadata is the essential blueprint for what data a company has, what it means and where they can get it. Without clear accessible metadata, an enterprise could have all the data they need to make sound decisions sitting on their servers and hard drives, and still end up making a best guess because they can't find that data or making decisions that they later regret because they only found a fraction of the data they really needed, or couldn't make sense of what they had. The trouble is, there's been a lot of talk for quite a while now about the need for and value of creating a warehouse for metadata that makes this critical information easily accessible, but businesses have spent too much money and seen far too little results

Traditional approaches to metadata management have been ad hoc at best. Simple oral knowledge transfer was not uncommon. In the past, database administrators or IT professionals would abstract metadata from data sources by hand, using Excel spreadsheets or even doc files to document data properties and relationships. This type of documentation wasn't exactly designed for answering queries, and locating these documents could be a challenge in itself since their location was generally undocumented. But, the biggest shortcoming of this pile-of-documents approach was that it had no strategy for dealing with the natural drift and change of data structures, value sets and relationships. Even assuming that the Excel spreadsheets or docs were accurate in the beginning, they would become increasingly inaccurate over time. It would be rare for overworked database administrators to consider it their job to keep all of those scattered spreadsheets up to date.The other option was to pay a million dollars or so to a consulting company or tool vendor for an ambitious multiyear project. For large sums of money, a vendor would cheerfully build a slick metadata repository with a great interface that provided a sharp, clear window into some fraction of the essential metadata that you had two years ago. Not quite what you need to make dynamic, adaptive decisions for this year, or this quarter.

Now that we have begun to move out of the dark ages of metadata management, various tool

Corresponding author: Dr.Ch .GVN Prasad

vendors across the enterprise are attempting to solve the metadata puzzle by storing the metadata that is relevant to the type of problem that tool is designed to solve. This is definitely a step in the right direction, but it's far from ideal. Multiple isolated silos of metadata create redundancy, inconsistency, and incompleteness of metadata across the enterprise. The matter is complicated by the sheer volume of data regularly processed in businesses today and the exponentially increasing number of new data and metadata sourcesIn one place, data profiling and quality tools store statistics about the anomalies in the data, value ranges and other valuable information. In another, integration tools store design-time information about source and target data structures, transformation lineage, business rules and definitions and process flow as well as run time information about production lineage, which versions were executed, and success, failure and error logs. In still other locations, database management systems—either stand-alone or embedded in various applications—store field rules, names, types, and table structures, relationships and interdependencies. Each tool stores metadata in its own proprietary format. And of course, there are still many applications that don't store their metadata as an entity at all, and require metadata to be extrapolated either by hand or by custom coding.If an important big picture inquiry or an in-depth technical question requires more than one of these types of metadata, there is no common place to go for the information. The information is scattered across multiple metadata repositories in multiple locations—and multiple storage methods rendering the accessibility of cross-functional metadata an unavailable option. On top of that, these tools and metadata storage mediums are part of the constantly changing business landscape. In today's world of continual mergers and acquisitions, changing business initiatives, and constantly increasing variety of applications, sources of both data and metadata are unstable, moving targets.The one lesson that both the hand documentation and the overpriced, insufficient metadata solutions of the past teach us is that capturing metadata—while it may seem like a daunting task is relatively easy. Modern tools have made that aspect of the job even easier. Achieving a unified view of all metadata across the enterprise without bankrupting the company, and keeping that metadata synchronized with the actual properties of the data sources, are the real challenges.

Metadata Management Strategy Checklist

The secret to meeting those challenges is to build the metadata solution on top of a versatile integration platform that exposes its own metadata for consolidation, automatically extrapolates metadata from sources that don't provide it, and provides the connectivity capability to extract metadata from a wide variety of data sources. Modern integration toolsets frequently include metadata management capabilities as part of the package. Since a good metadata strategy requires bringing together a huge variety of disparate data sources and since a fair number of the benefits of metadata management are directly related to data and application integration projects, it makes sense to not just build the metadata warehouse with integration processes, but to build the metadata management into the warehouse and processes Consolidate Metadata

1. The first step in a good overall metadata strategy is to extract the
metadata out of its isolated silos and bring it all together. This will reduce data redundancy, duplication and inconsistencies. This is, in its essence, an exact-transformation-load (ETL) problem; using a good ETL tool to get the metadata out of various repositories, cleanse and aggregate it and load it into a metadata warehouse makes perfect sense.

2. Automate Synchronization

A scheduled or change-event driven automated integration
process can make certain that the metadata warehouse is regularly updated and will remain synchronized over time with the changing sources, without adding to anyone's ongoing workload. In addition, when future data sources are added, with an integration toolset that has a decent user interface, it's a simple matter to extend already established integration processes to include new data sources.

Make ALL Enterprise Metadata Easily Searchable - A comprehensive metadata warehouse with a well-documented star schema or other query optimized structur
e, can make a world of difference in terms of the speed and quality of answer. The data warehouse

should be designed for optimal use with your existing or desired business intelligence (BI) reporting tools, but not be limited to only BI-related metadata. It should enable you to retrieve all the metadata that matters to everyone across the enterprise, whether business or technical, front office or back office, into a single location, and then provide multiple windows into that data. This will provide a clear global view of the enterprise and enable knowledge transfer and information sharing. It also means that the metadata that matters to any particular user will be easy to find when needed. You will be able to find answers to technical, business, or cross-boundary issues that haven't even been conceived of yet in a good metadata warehouse.

4. Get your ROI perspective in shape- So, you've got a good idea of how to build an enterprise-wide robust metadata management strategy. The question now is, "Why bother?" It's tough to show the ROI for metadata initiatives in hard numbers up front, especially since a lot of the benefits show up over a fairly significant length of time. It's essential to have a low-cost option that makes ROI a no-brainer. A low cost point can make the difference between months of justification, and a low effort, back-of-an-envelope kind of calculation. Start tactically. Build a small, viable data warehouse that includes just the information for a particular business segment, such as a division or department. This allows the demonstration of clear ROI with low risk. It also gives you a chance to build model integration and synchronization processes that can then be re-used or extended to other departments, at an even lower cost.

A solid metadata strategy can show a clear increase in ROI and decrease in cost point for a variety of other important initiatives as well. For instance, it can help to show adherence to data governance and compliance requirements by providing a clear audit trail.

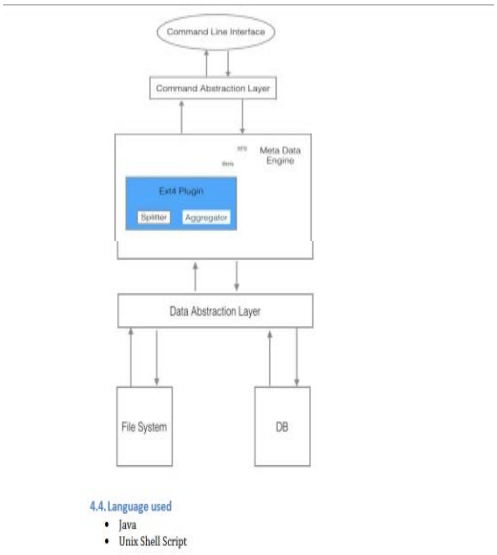
Benefits of Robust Metadata Management

Metadata management is critical for organizations that rely on multiple data sources for business intelligence, especially if some of those data sources are older, or less easily accessible. Including even hard-to-get metadata improves the knowledge base available for BI queries, giving more relevant, accurate and useful analyses and reporting. The ability to view and analyze both technical and business metadata also provides a mechanism that ensures that the value being extracted from the data continues to meet business objectives. This can improve IT decision-making by validating technical processes with business goals.For integration projects, metadata provides data lineage throughout various stages of transformation for future error checking, compliance auditing, and data quality improvements. Good metadata can help with any business policy or infrastructure change by providing information that helps companies gauge the complexity of changes and plan the best use of resources. This can significantly reduce development and maintenance costs. Management of integration design metadata can also provide the basis for module re-use. Enabling collaboration and component re-use significantly accelerates project timelines. The creatie use of metadata can provide innovative approaches to long-standing integration problems that provide completely unforeseen benefits. A good example is the Vision Award-winning process of LifeMasters Supported SelfCare, Inc. Using metadata on top of a versatile data integration platform, LifeMasters was able to reduce patient data processing time from five days to two hours, and new patient on-ramping from 7-10 days down to 1-2 days.

Overall, the goal of a metadata management strategy is to reduce IT costs and increase corporate productivity and agility, and that always translates one way or another to increased ROI. With a solid, broad-spectrum, easily searchable metadata warehouse and automated updating processes in place to keep it current, an enterprise will undoubtedly see benefits, including those the organization may not have thought of. Executives, data analysts and developers will also genuinely understand their data descriptions, definitions, lineage and relationships. The metadata that really matters will be at their fingertips.

Metadata Query Language

findattr <ATTRNAME>[AND OR] <ATTRNAME>[AND OR]...



libFUSE
o
Filesystem in Userspace
libraries that allows
to implement a fully
functional filesystem in a userspace program
•
fuse-jna
o
fuse-jna is a FUSE API library that u
ses Java Native Access to
communicate with the libFuse
•
Redis NoSql Database
o
Redis is an open source, BSD licensed, advanced
key-value cacheand store

Creating files:
In order to load test, the below program creates the files within the Mirror File System.
To Speed up the operations, the program is multithreaded.
Create 100000 files in 10 Threads

```
java -cp build/libs/\* coen283.p3.CreateFiles fstest
target/single-thread-files 100000 10
```

Adding Metadata:

OSX:
Using '**xattr**' command

```
# xattr -w <attribute-name> <attribute-value> <filename>
```

Linux:

Using '**setfattr**' command

```
# setfattr -n <KEY> -v "<VALUE>" <FILE>
```

Implementation
1. Create 100,000 files on Native File System
2. Create 100,000 files on Metadata File System
3. Measure Time taken for Adding attributes to 10,000 files
4. Measure time taken for retrieval of attributes from 10,000 files.
The System Parameters used for testing are OSX
–
10.10/2.6 GHz Intel Core i5 / 16GB RAM 1600 MHz DDR3/ 128GB SSD Linux
Ubuntu 14.14 / Virtual Machine / 8GB RAM / 20GB Virtual HDD
5. Implementation Code (refer programming requirements)
The Program is submit ed using 'Submit' script as P3 project.
The Program can be run on a linux system with the following packages installed
1. libfuse2.redis-server3.openjdk-7 or Oracle JDK 74.fattr
Command to process extended arguments 'Autotest' and 'stdin' would not apply to the project as the Design Center Lab workstations do not support libfuse and redis database
Compiling Source Code:
The source code compiles using gradle wrapper. To compile execute the following command.

```
# ./gradlew rtJar
```

This generates a single JAR file with all the required libraries.

The jar file is generated at **build/libs/coen283-p3-t6.jar**

RUN the file system:

1. Create the mount point

```
# mkdir target/mirrorFS
```

2. Launch the filesystem

```
# ./run.sh
or
# java -cp build/libs/\* coen283.p3.MetadataFileSystem
```

Design document and flowchart

The following are the steps involved.

• The Command is given as an input inthe CLI

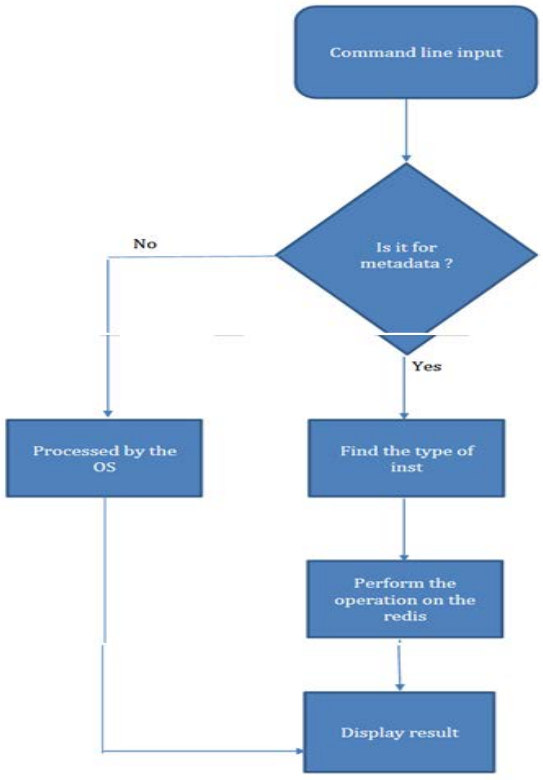
This command hits the MetaDatafilesystem which we have mounted using libFuse.

•Here we make a decision whether to redirectit tothe OS or the Metadat a Engine.

If it is Xattr based instruction then we go with the MetaData Engine.

•Based on the decisionif the option is MetaData Engine then we go to the module which performs the task for that particular command and display

•Repeat the Steps for the next command



Data analysis and discussion

Verifying from Command Line:

OSX:

Using

'xattr'

command.

```
# xattr -l target/mirrorFS/mf4*.txt
```

his displays the extended attributes that are added for all the files

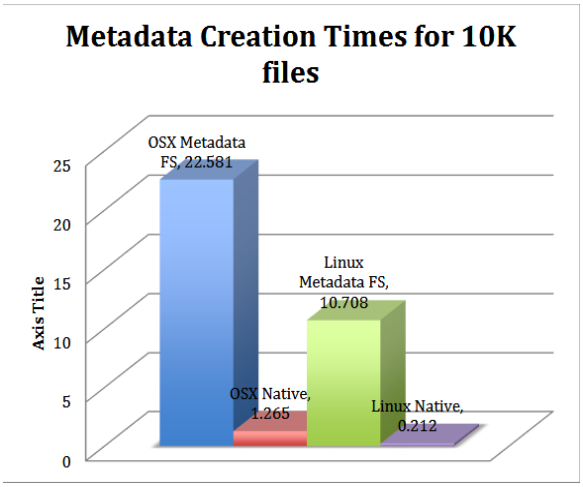
matching with 'f1*.txt'

Linux:

Using '

getfattr' command

```
# getfattr -n owner target/mirrorFS/mf4*
# getfattr -n purge-date target/mirrorFS/mf4*
```



The comparison of Metadata Retrieval times between Native Filesystems (OSX and Linux) and Metadata FileSystem is presented below. The data was measured for 10,000 files

Conclusion

We conclude that met adata management functionality into local filesystem such that it eliminates domain specific m etadata management applications from building its their own retrieval system . We can see that this works on both the OSx and also linux which implies that it is portable

We have seen that we have a long delay due to Fuse filesystem . So, we can check if we can implement it into the kernel space then the results will improve. The Metadata FileSystem can also be

used for a group of filesystems maintaining a single redix server

Bibliograph

1. ya.van Heuven van Staereling, Richard, et al. "Efficient, modular metadata management with loris." Networking, Architecture and Storage (NAS), 2011 6th IEEE International Conference on. IEEE, 2011.b.
2. Ren, Kai, and Garth Gibson. "TABLEFS: Embedding a NOSQL database inside the local file system." APMRC, 2012 Digest. IEEE, 2012.
3. C.Ren, Kai, and Garth A. Gibson. "TABLEFS: Enhancing Metadata Efficiency in the Local File System." USENIX Annual Technical Conference. 2013.