

# Notes

Quick note:

1. based on `Bugfix_entropy`, and distributed systems-related cases in `bugs.jar`;
2. focus on modified lines of patches, use `git blame` to find which line was last written/modified before, that line may be directly related to the bug. Try to learn patterns based on that line (hopefully content-based or dataflow-based).

Todos:

1. filter the `Bugfix_entropy` for bug-related examples;
2. find the last line that was written/modified and try to generate mutants by ML methods.

## Bugfix Entropy

### Existing Bug Repositories

1. Jira search term: `project = HADOOP AND resolved >= 2020-01-01 AND resolved < 2021-01-01 AND resolution = fixed AND type = bug ORDER BY created DESC`

Reporter not equal to assignee

Both fixed and resolved

Time duration

(Filter test)

It already set the type as "bug", but not enough for filtering, as there are many commits related to **importing packages, changing dependencies, adding files, syntax errors, switching branches, adding test cases**, etc. in the `Bugfix_entropy` repository.

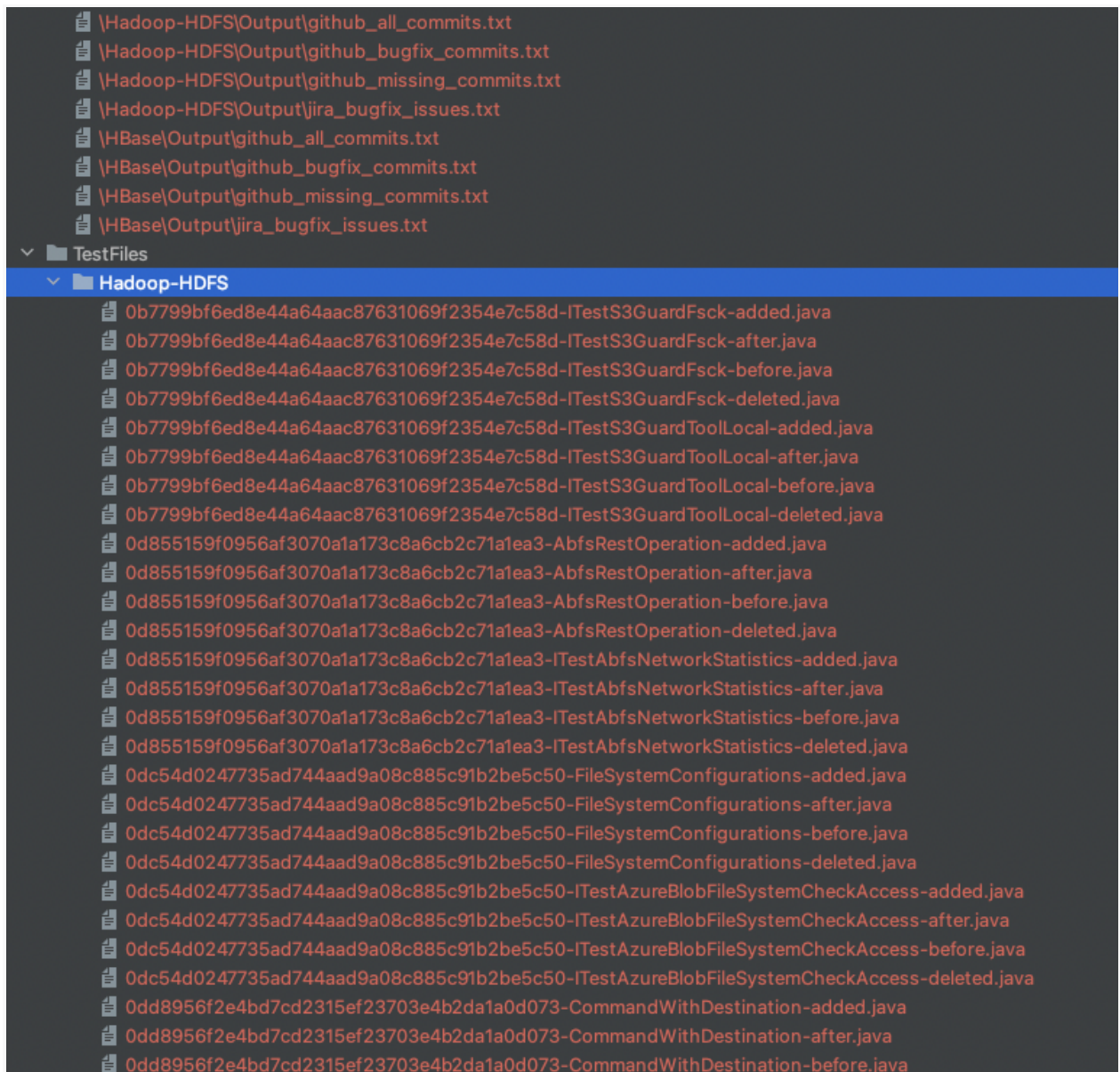
Shall we have another step to filter those commits before analyzing them?

2. Prints the files related to bugfix commits
  - 1) Numbers
    - a) Total GitHub Commits
    - b) Total Jira Bug Fixes
    - c) Total GitHub Bug Fixes

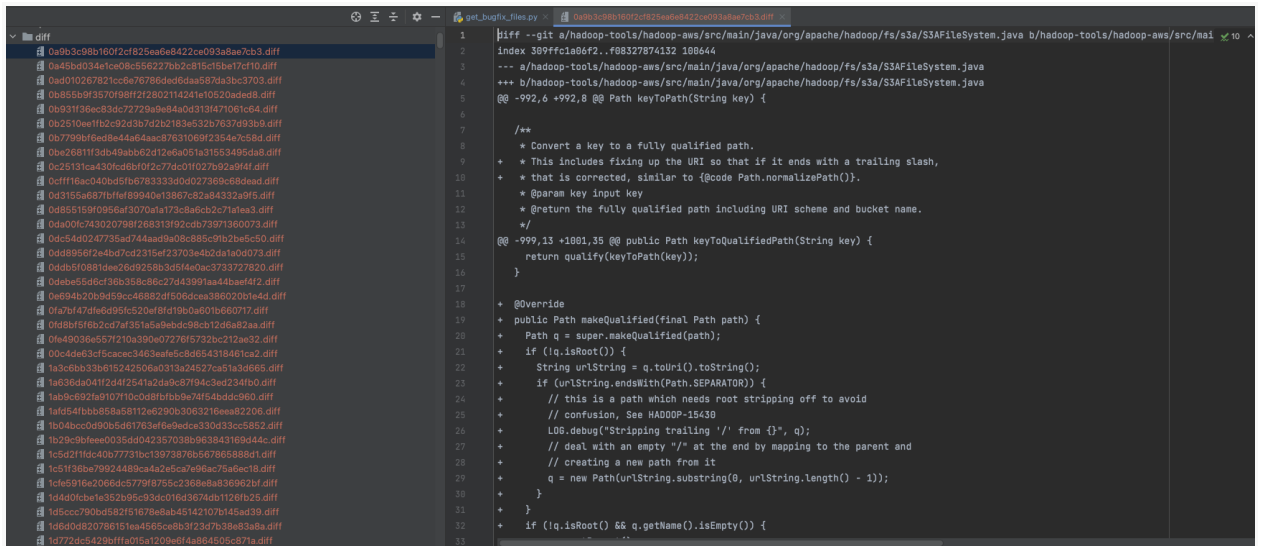
Those commits are not all directly related to the Jira tickets (only search commits by keywords), maybe that is why many commits are not fixing bugs.  
How to filter them?

```
get_bugfix_files x
/Users/moeachto/.conda/envs/envs/bin/python /Users/moeachto,
[2022-10-18 14:08:42.636701] -----
[2022-10-18 14:08:42.636966] Hadoop-HDFS
[2022-10-18 14:09:13.166012] Total GitHub Commits: 1104
[2022-10-18 14:09:13.166059] Total Jira Bug Fixes: 132
[2022-10-18 14:09:13.166067] Total GitHub Bug Fixes: 108
[2022-10-18 14:09:13.166277] -----
[2022-10-18 14:09:13.166290] HBase
[2022-10-18 14:09:42.995523] Total GitHub Commits: 1146
[2022-10-18 14:09:42.995559] Total Jira Bug Fixes: 355
[2022-10-18 14:09:42.995567] Total GitHub Bug Fixes: 309
Run time: 0:01:00.359667
```

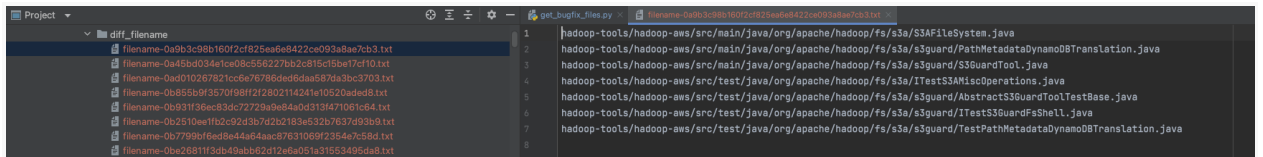
2) Snapshots of the **added, deleted, before, and after** codes for a commit



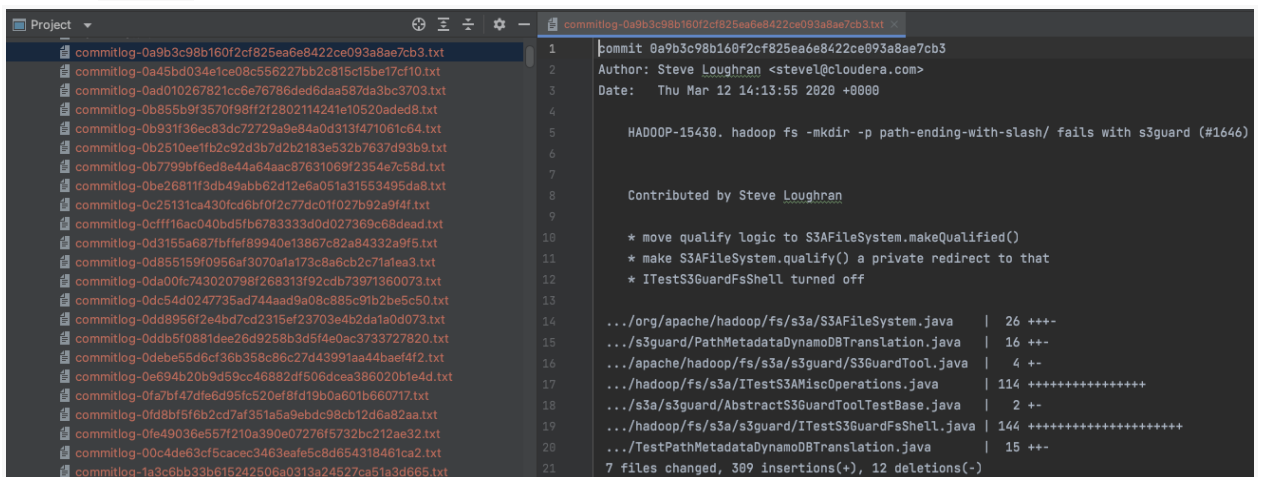
### 3) All diff lines of commits



### 4) All modified file names of commits (save for git blame)



### 5) Commit logs (show number of modified lines)



## 3. git blame <filename> <commit-hash>

Unsolved difficulties:

### 1) Modifying can trace back the latest commit with time, but what about adding?

## Focus on control/data flow

```
074050ca595a (Takanobu Asanuma) 2020-01-01 11:26:38 +0900 160) public static Map<String, Object> getEcPolicyAsMap(
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 161)     final ErasureCodingPolicy ecPolicy) {
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 162)     /** Convert an ErasureCodingPolicy to a map. */
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 163)     ImmutableMap.Builder<String, Object> builder = ImmutableMap.builder();
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 164)     builder.put("name", ecPolicy.getName())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 165)         .put("cellSize", ecPolicy.getCellSize())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 166)         .put("numDataUnits", ecPolicy.getNumDataUnits())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 167)         .put("numParityUnits", ecPolicy.getNumParityUnits())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 168)         .put("codecName", ecPolicy.getCodecName())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 169)         .put("id", ecPolicy.getId())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 170)         .put("extraOptions", ecPolicy.getSchema().getExtraOptions());
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 171)     return builder.build();
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 172) }
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 173) }

0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 160) private static Map<String, Object> getEcPolicyAsMap(
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 161)     final ErasureCodingPolicy ecPolicy) {
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 162)     /** Convert an ErasureCodingPolicy to a map. */
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 163)     ImmutableMap.Builder<String, Object> builder = ImmutableMap.builder();
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 164)     builder.put("name", ecPolicy.getName())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 165)         .put("cellSize", ecPolicy.getCellSize())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 166)         .put("numDataUnits", ecPolicy.getNumDataUnits())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 167)         .put("numParityUnits", ecPolicy.getNumParityUnits())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 168)         .put("codecName", ecPolicy.getCodecName())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 169)         .put("id", ecPolicy.getId())
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 170)         .put("extraOptions", ecPolicy.getSchema().getExtraOptions());
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 171)     return builder.build();
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 172) }
0fc988e6a3dc (Arpit Agarwal) 2018-05-16 11:28:39 -0700 173) }
```

## 2) How to show only the diff lines of a commit when using git blame?

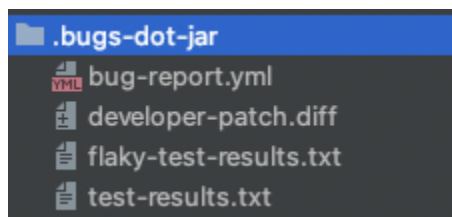
Not finished yet. May require further scripting, or parse outputs

# Bugs.jar

Table 1: Selected subject systems

Project	Purpose	Tags	Commits	Bug Reports	Size (excl. tests) [KLoC]	Total Size [KLoC]	Bugs Selected
Accumulo	sorted, distributed key-value store	database	8,714	2,041	371	458	98
Camel	routing and mediation engine	network-client, network-server	24,096	1,081	122	257	147
Commons Math	math & statistics library	library	5,994	635	93	187	147
Flink	streaming dataflow engine	big data	8,906	2,070	171	345	70
Jackrabbit Oak	content management system	XML, network-server, library	10,810	1,686	139	228	278
Log4J2	logging framework	library	6,971	784	63	104	81
Maven	project management	build management	10,264	2,863	81	100	48
Wicket	server-side Web app framework	Web framework	19,386	3,770	127	177	289
<b>Total</b>			<b>95,141</b>	<b>14,930</b>	<b>1,167</b>	<b>1,856</b>	<b>1,158</b>

1. 8 selected Apache projects  
Select projects directly related to distributed systems.
2. Identify bug-fixing commits by Jira type as Bug, and extract the version named V\_fix  
This step may not be reliable enough, as there are many commits related to importing packages, changing dependencies, adding files, syntax errors, switching branches, adding test cases, etc. in the Bugfix\_entropy repository.
3. Apply the bug-fixing reverse patch to get the buggy version of the source code named V\_buggy  
If trying to reintroduce a new bug, then it is the same idea as FIXREVERTER
4. Run the test cases on V\_buggy. If there is any fault-reproducing test case, then consider the bug for further investigation. Run the previous step 10 times to ensure the bugs are consistently reproducible.  
Not all bugs are reproducible. Is there a more reasonable/advanced principle for building our own corpus?
5. Manually analyze each one of a set of consistently reproducible bugs to make sure that they are indeed a bug



Single-line example:

```
diff --git a/core/src/main/java/org/apache/accumulo/core/iterators/Combiner.java  
b/core/src/main/java/org/apache/accumulo/core/iterators/Combiner.java
```

```

index 6e72073..584eb14 100644
--- a/core/src/main/java/org/apache/accumulo/core/iterators/Combiner.java
+++ b/core/src/main/java/org/apache/accumulo/core/iterators/Combiner.java
@@ -63,7 +63,7 @@ public abstract class Combiner extends WrappingIterator
implements OptionDescrib
    */
    public ValueIterator(SortedKeyValueIterator<Key,Value> source) {
        this.source = source;
-        topKey = source.getTopKey();
+        topKey = new Key(source.getTopKey());
        hasNext = _hasNext();
    }

```

Multi-lines example:

```

diff --git
a/core/src/main/java/org/apache/accumulo/core/security/Authorizations.java
b/core/src/main/java/org/apache/accumulo/core/security/Authorizations.java
index 5933325..a677f3f 100644
--- a/core/src/main/java/org/apache/accumulo/core/security/Authorizations.java
+++ b/core/src/main/java/org/apache/accumulo/core/security/Authorizations.java
@@ -23,10 +23,9 @@ import java.nio.charset.Charset;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
+import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
-import java.util.Set;
-import java.util.TreeSet;

import org.apache.accumulo.core.data.ArrayByteSequence;
import org.apache.accumulo.core.data.ByteSequence;
@@ -38,14 +37,14 @@ public class Authorizations implements Iterable<byte[]>,
Serializable {

    private static final long serialVersionUID = 1L;

-    private Set<ByteSequence> auths = new TreeSet<ByteSequence>();
+    private HashSet<ByteSequence> auths = new HashSet<ByteSequence>();
    private List<byte[]> authsList = new ArrayList<byte[]>();
    private List<byte[]> immutableList = Collections.unmodifiableList(authsList);

    private static final boolean[] validAuthChars = new boolean[256];

```



```

    public static final String HEADER = "!AUTH1:";
-
+
    static {
        for (int i = 0; i < 256; i++) {
            validAuthChars[i] = false;
@@ -104,11 +103,11 @@ public class Authorizations implements Iterable<byte[]>,
Serializable {
    * @param authorizations
    *         a serialized authorizations string produced by {@link
#getAuthorizationsArray()} or {@link #serialize()}
    */
-
+
    public Authorizations(byte[] authorizations) {

        ArgumentChecker.notNull(authorizations);
-
+
        String authsString = new String(authorizations);
        if (authsString.startsWith(HEADER)) {
            // its the new format
@@ -141,7 +140,7 @@ public class Authorizations implements Iterable<byte[]>,
Serializable {
    public Authorizations(Charset charset, String... authorizations) {
        setAuthorizations(charset, authorizations);
    }
-
+
    public Authorizations(String... authorizations) {
        setAuthorizations(authorizations);
    }
@@ -177,7 +176,6 @@ public class Authorizations implements Iterable<byte[]>,
Serializable {
    return ByteBufferUtil.toByteBuffers(immutableList);
}

- @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        String sep = "";
@@ -198,7 +196,6 @@ public class Authorizations implements Iterable<byte[]>,
Serializable {
    return auths.contains(auth);
}

- @Override

```



```

    public boolean equals(Object o) {
        if (o == null) {
            return false;
        }
    }

    @@ -213,7 +210,6 @@ public class Authorizations implements Iterable<byte[]>,
    Serializable {
        return false;
    }

-   @Override
    public int hashCode() {
        int result = 0;
        for (ByteSequence b : auths)

```