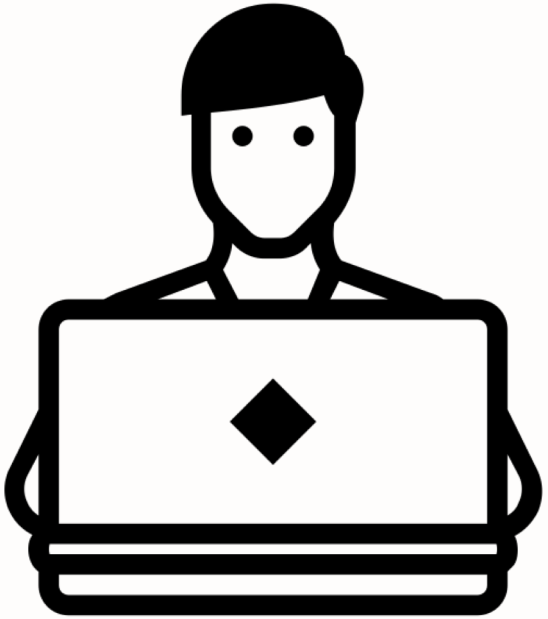
MSR 2018

Toward Predicting Architectural Significance of Implementation Issues

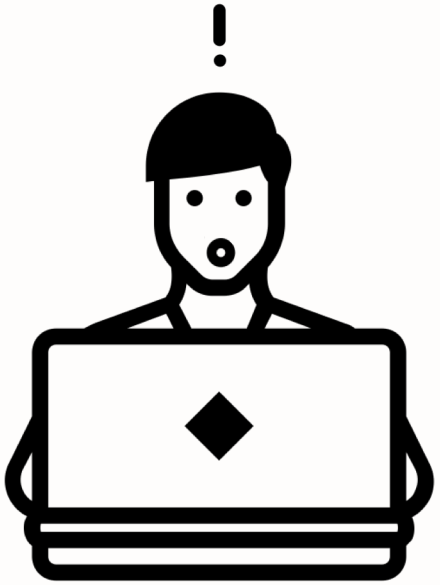
Arman Shahbazian, Daye Nam, and Nenad Medvidovic

University of Southern California

Motivation



Motivation



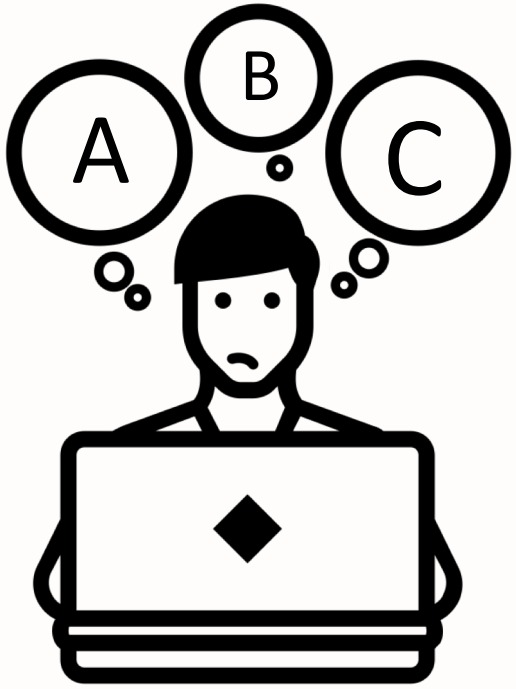
Very Critical Urgent Issue

#18065 opened 28 minutes ago by user1

blocker performance bug

help wanted server hotfix

Motivation



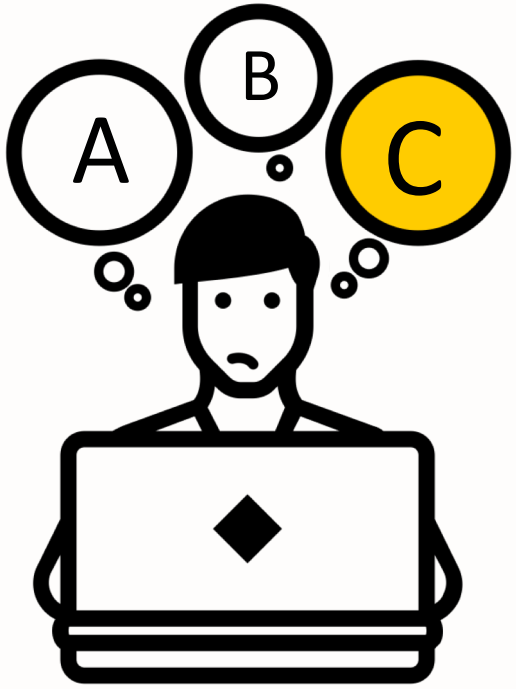
Very Critical Urgent Issue

#18065 opened 28 minutes ago by user1

blocker performance bug

help wanted server hotfix

Motivation



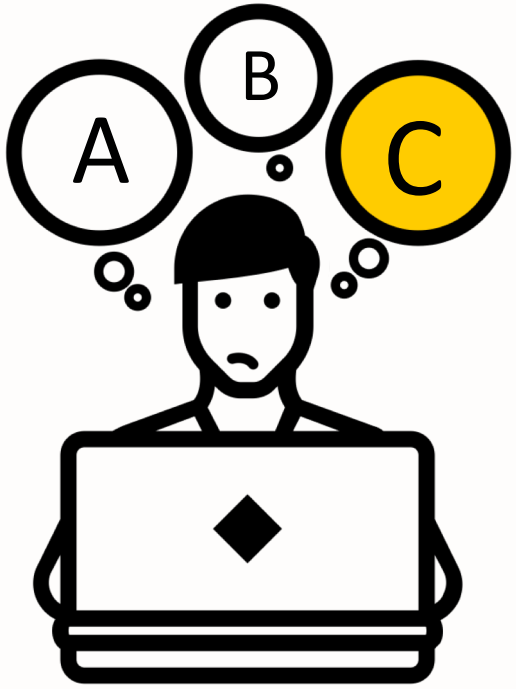
Very Critical Urgent Issue

#18065 opened 28 minutes ago by user1

blocker performance bug

help wanted server hotfix

Motivation



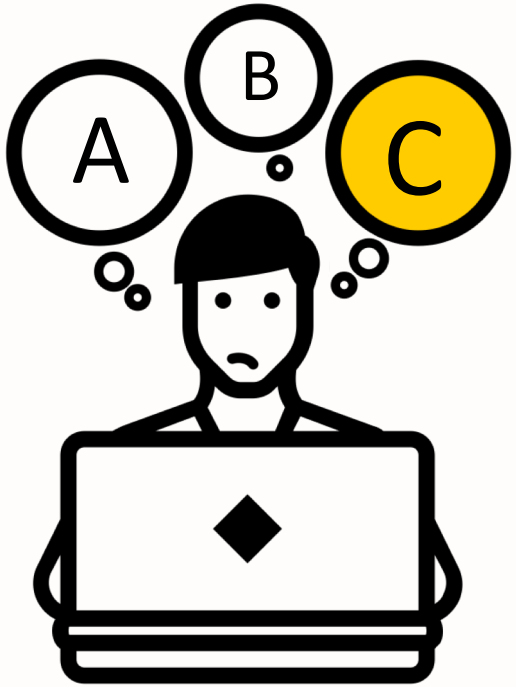
Very Critical Urgent Issue

#18065 opened 28 minutes ago by user1

blocker performance bug

help wanted server hotfix

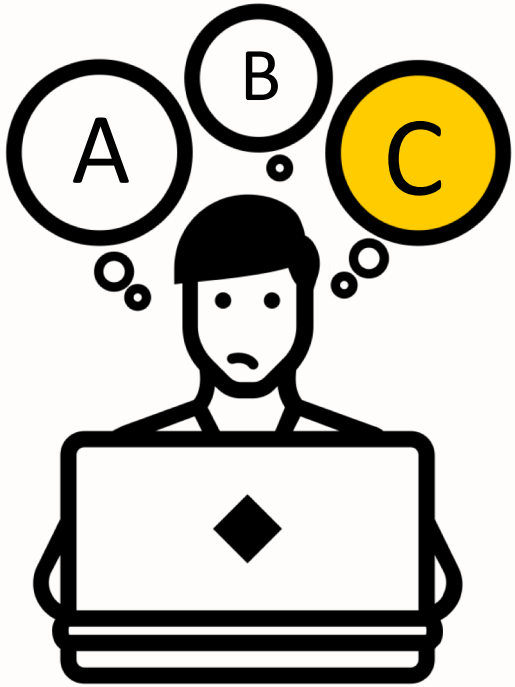
Motivation



Numerous Design Decisions

Inadvertent Architectural Changes

Motivation

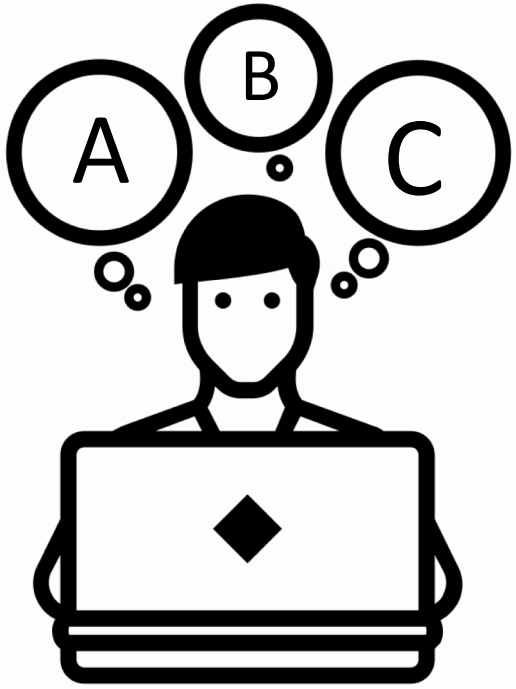


Numerous Design Decisions

Inadvertent Architectural Changes

- **Accumulation of Technical Debt**
- **Deterioration of Software Quality**

Motivation



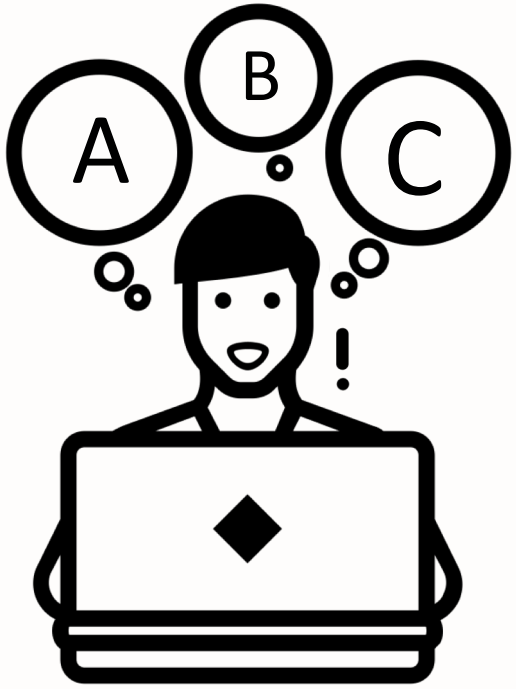
Very Critical Urgent Issue

#18065 opened 28 minutes ago by user1

blocker performance bug

help wanted server hotfix

Motivation



Architecturally Significant

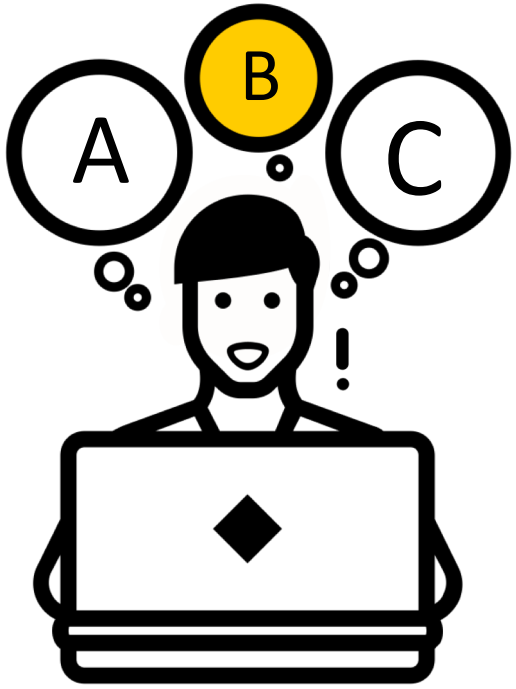
Very Critical Urgent Issue

#18065 opened 28 minutes ago by user1

blocker performance bug

help wanted server hotfix

Motivation



Architecturally Significant

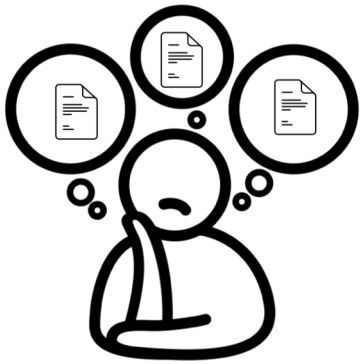
Very Critical Urgent Issue

#18065 opened 28 minutes ago by user1

blocker performance bug

help wanted server hotfix

Contributions

Detection	Dataset	Classifier
		

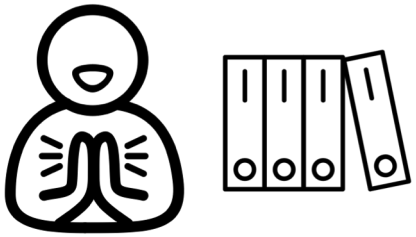
Contributions

Detection

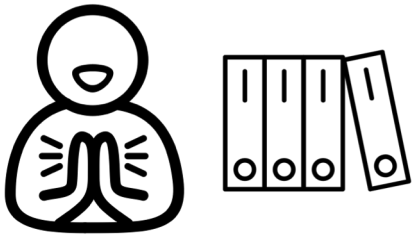
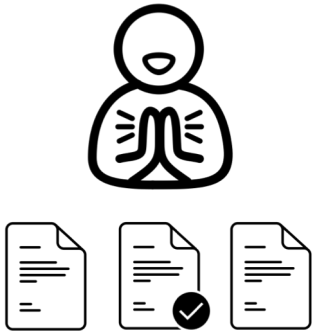
Dataset

Classifier

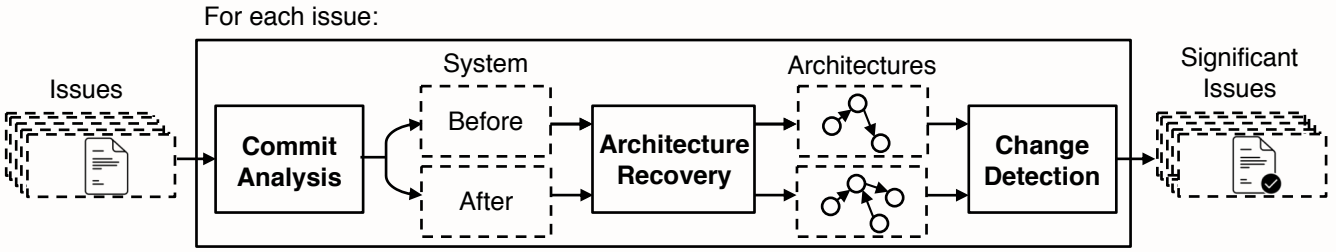
Contributions

Detection	Dataset	Classifier
		

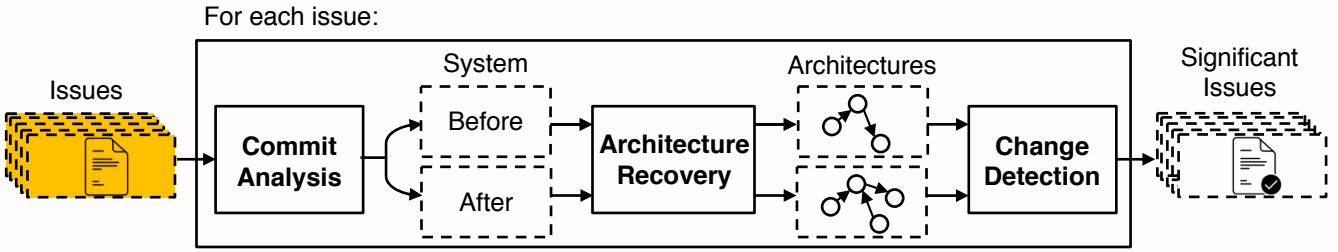
Contributions

Detection	Dataset	Classifier
		

Detection + Dataset



Detection + Dataset



OpenJPA

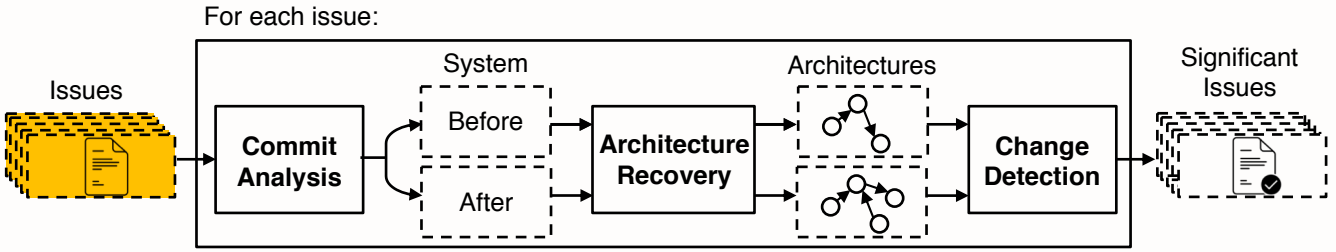
APACHE WICKET

nutch

Apache CXF

hadoop

Detection + Dataset



Hadoop Common / HADOOP-1096

Rename InputArchive and OutputArchive and make them public

Details

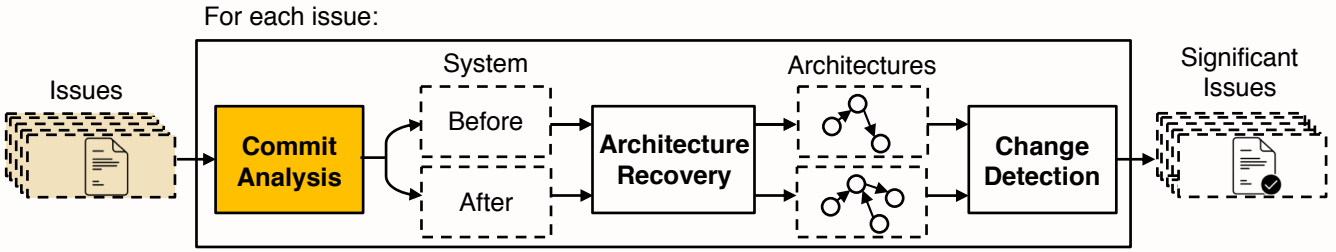
Type:	Improvement	Status:	CLOSED
Priority:	Major	Resolution:	Fixed
Affects Version/s:	0.12.0	Fix Version/s:	0.12.1
Component/s:	record		
Labels:	None		
Environment:	All		

Description

Currently hadoop.record.RecordReader and RecordWriter act as factories for various InputArchive and OutputArchive recently. In the original design, this was done in order to have tight control over various serialization formats. This has proven to be counterproductive. For wider usage of record I/O one should be able to use their own serialization formats. The proposed changes make it possible. They are as follows:

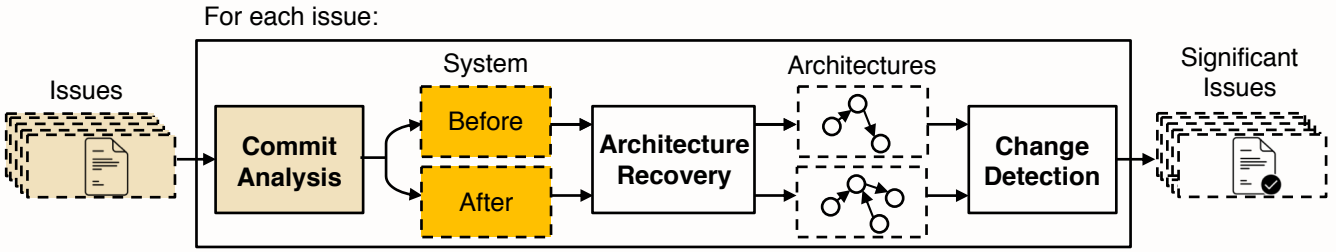
1. Eliminate current record.RecordReader and record.RecordWriter.
2. rename InputArchive as RecordInput, and OutputArchive as RecordOutput.
3. rename various archives accordingly. e..g. BinaryInputArchive -> BinaryRecordInput etc.

Detection + Dataset




```
Index: src/java/org/apache/hadoop/record/InputArchive.java
===================================================================
--- src/java/org/apache/hadoop/record/InputArchive.java (revision 519069)
+++ src/java/org/apache/hadoop/record/InputArchive.java (working copy)
@@ -1,44 +0,0 @@
```

Detection + Dataset



Hadoop Common / HADOOP-1096

Rename InputArchive and OutputArchive and make them public

Details

Type:	Improvement	Status:	CLOSED
Priority:	Major	Resolution:	Fixed
Affects Version/s:	0.12.0	Fix Version/s:	0.12.1
Component/s:	record		
Labels:	None		
Environment:	All		

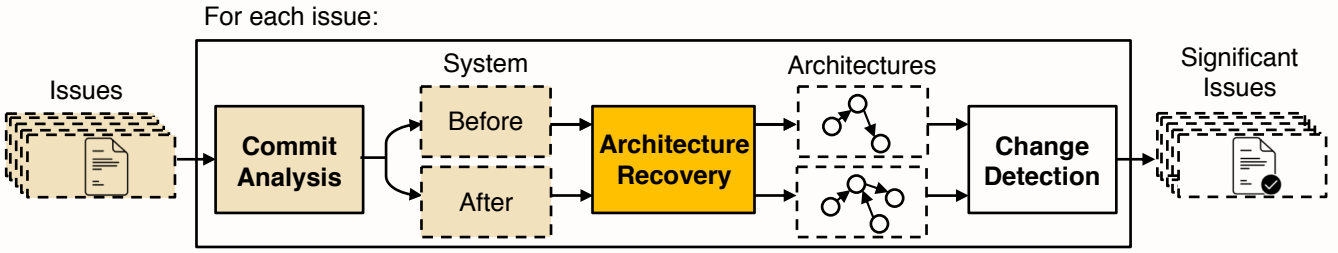
0.12.0

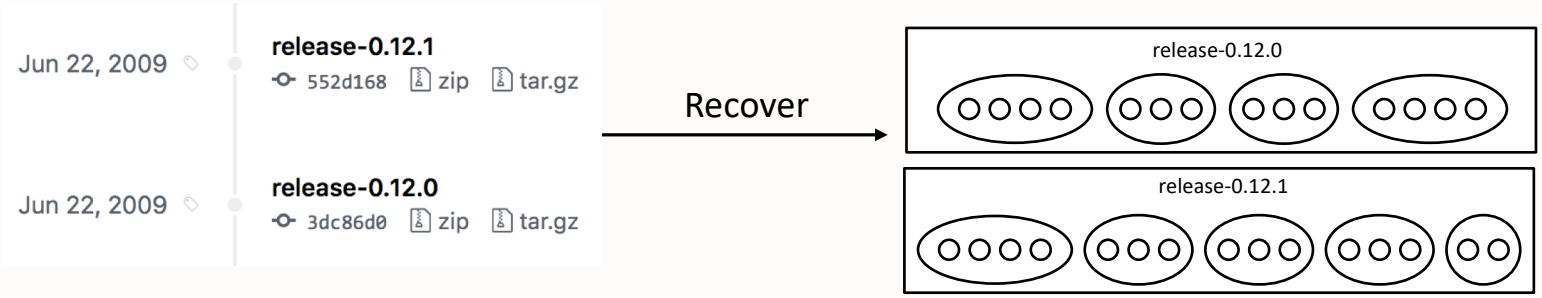
0.12.1

Description

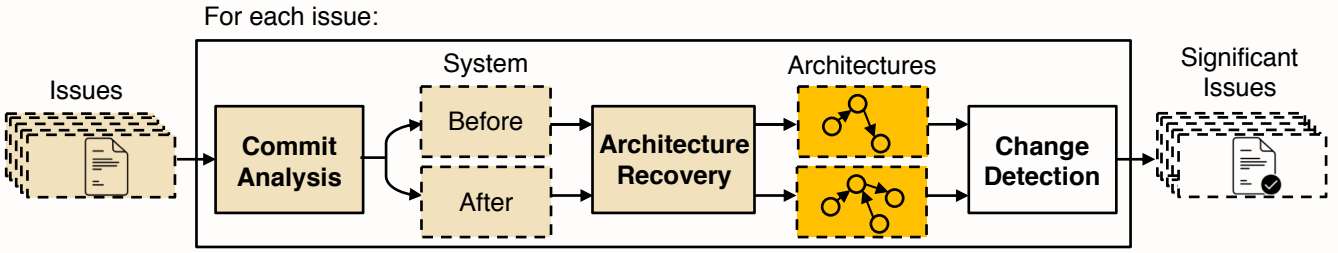
Currently hadoop.record.RecordReader and RecordWriter act as factories for various InputArchive and OutputArchive recently. In the original design, this was done in order to have tight control over various serialization formats. This has proven to be counterproductive. For wider usage of record I/O one should be able to use their

Detection + Dataset

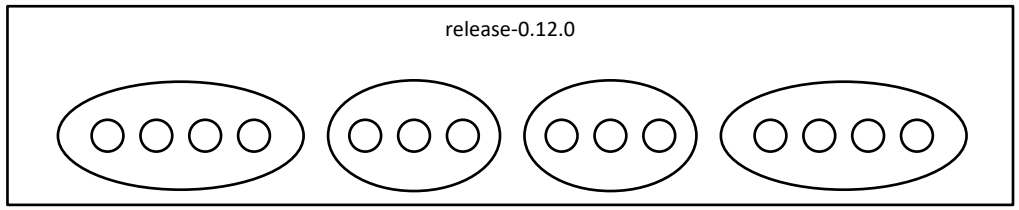




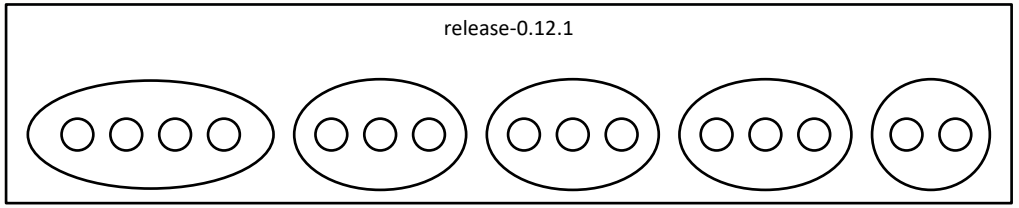
Detection + Dataset



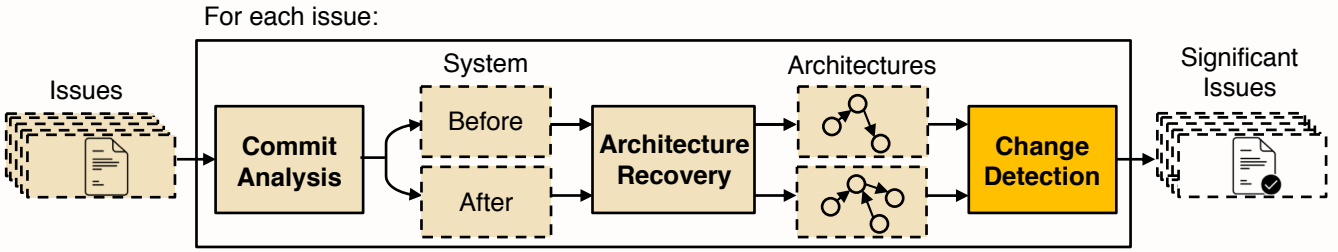
Before

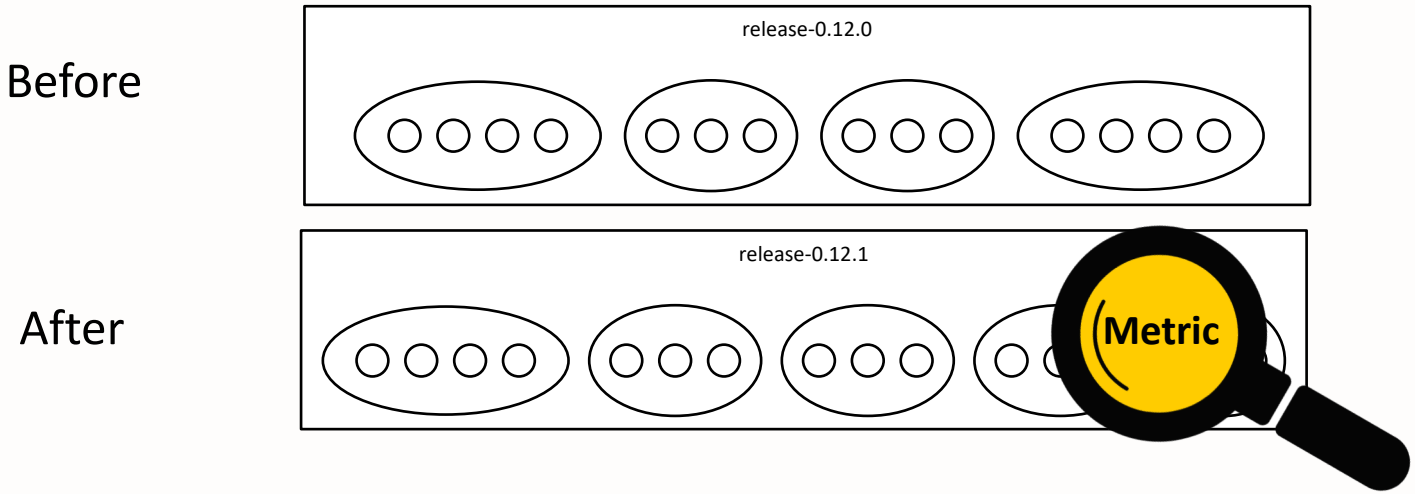


After

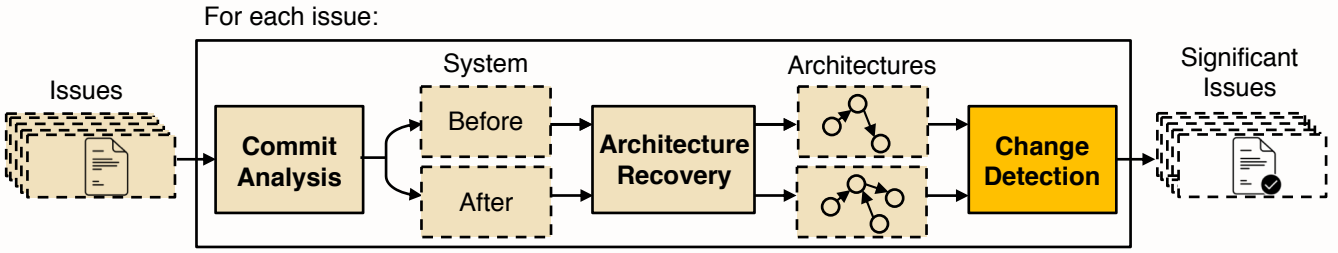


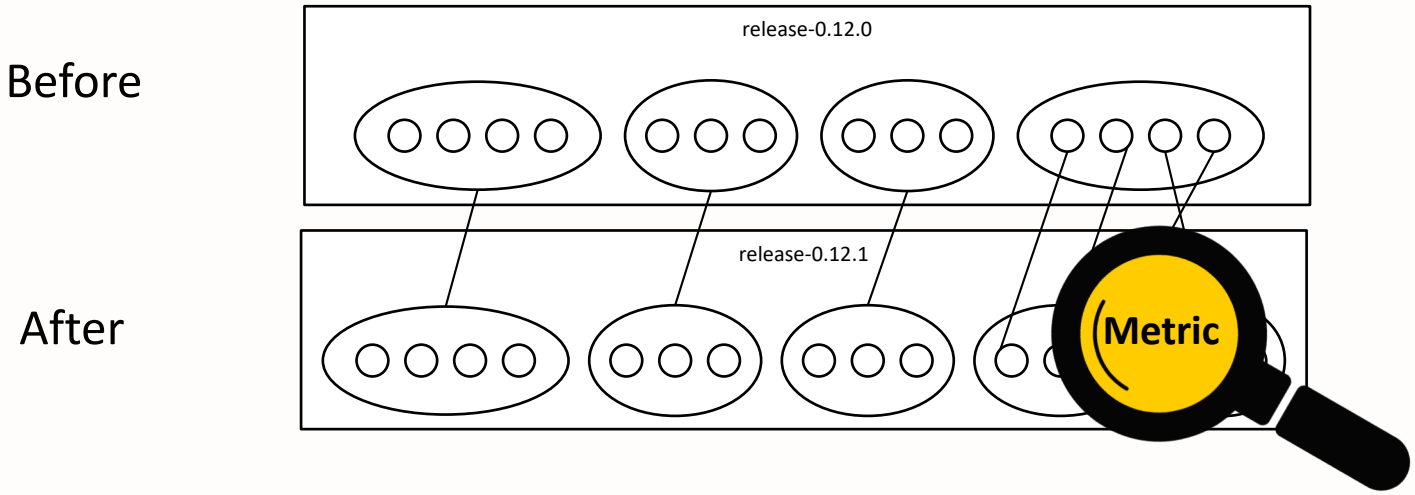
Detection + Dataset



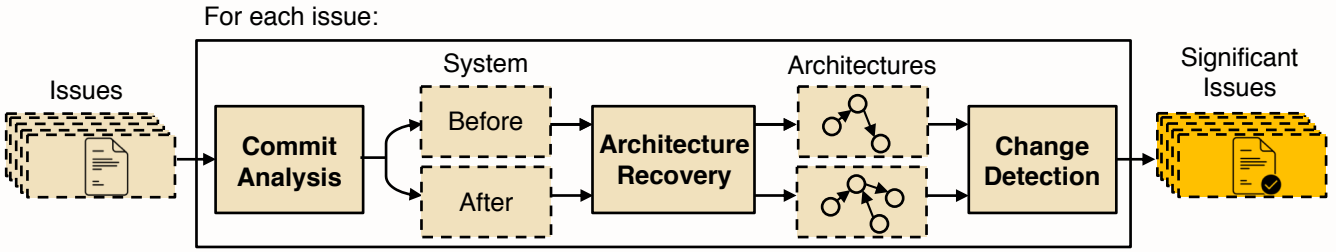


Detection + Dataset





Detection + Dataset



Hadoop Common / HADOOP-1096

Rename InputArchive and OutputArchive and make them public

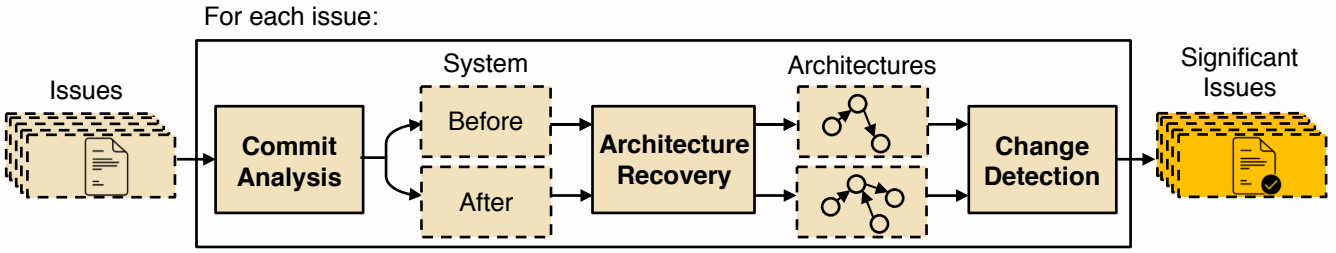
Details

Type:	Improvement	Status:	CLOSED
Priority:	Major	Resolution:	Fixed
Affects Version/s:	0.12.0	Fix Version/s:	0.12.1
Component/s:	record		
Labels:	None		
Environment:	All		

Description

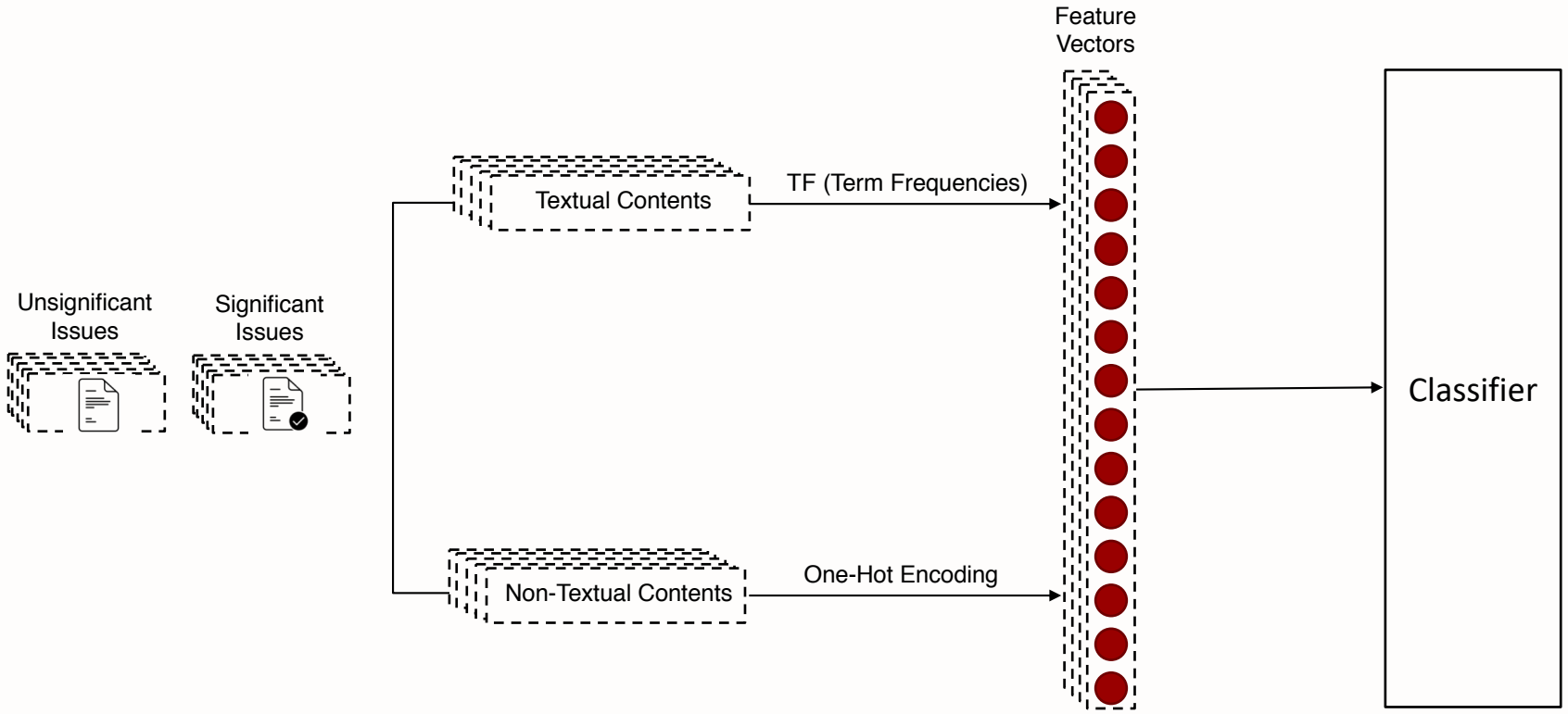
Currently hadoop.record.RecordReader and RecordWriter act as factories for various InputArchive and OutputArchive recently. In the original design, this was done in order to have tight control over various serialization formats. This has proven to be counterproductive. For wider usage of record I/O one should be able to use their

Detection + Dataset



https://softarch.usc.edu/predictar

Contributions



Evaluation

	Precision	Recall
Hadoop	0.838	0.592
Nutch	0.946	0.247
Wicket	0.761	0.537
Cxf	0.865	0.538
OpenJpa	0.934	0.451
Cross-Project	0.811	0.583

Evaluation

	Precision	Recall
Hadoop	0.838	0.592
Nutch	0.946	0.247
Wicket	0.761	0.537
Cxf	0.865	0.538
OpenJpa	0.934	0.451
Cross-Project	0.811	0.583

Evaluation

	Precision	Recall
Hadoop	0.838	0.592
Nutch	0.946	0.247
Wicket	0.761	0.537
Cxf	0.865	0.538
OpenJpa	0.934	0.451
Cross-Project	0.811	0.583

Evaluation

	Precision	Recall
Hadoop	0.838	0.592
Nutch	0.946	0.247
Wicket	0.761	0.537
Cxf	0.865	0.538
OpenJpa	0.934	0.451
Cross-Project	0.811	0.583

Conclusion

Summary

- Automatically detecting architecturally significant issues,
- A reusable dataset of over 21,000 issues,
- Classifying them based on information contained in each issue.

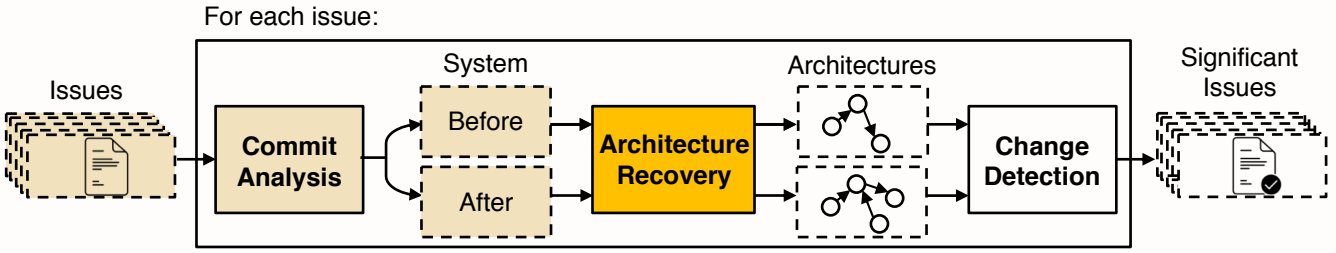
Future Work

- Expand to more systems by adding the support for other issue trackers,
- Improve the performance by adding more data,
- Improve the performance by adapting new model in Machine Learning.

THANK YOU

(armansha@usc.edu, dayenam@usc.edu, neno@usc.edu)

Data Collection



- ACDC: Algorithm for Comprehension-Driven Clustering
  - Structural pattern-based clustering
- ARC: Architecture Recovery using Concerns
  - Concern-based hierarchical clustering based on similarity measure

Evaluation

	ARC		ACDC	
	Precision	Recall	Precision	Recall
Hadoop	0.793	0.637	0.883	0.547
Nutch	0.941	0.276	0.951	0.217
Wicket	0.843	0.657	0.678	0.417
Cxf	0.801	0.698	0.928	0.468
OpenJpa	0.965	0.503	0.903	0.399
Cross-Project	0.816	0.592	0.806	0.573

Evaluation

	ARC		ACDC	
	Precision	Recall	Precision	Recall
Hadoop	0.793	0.637	0.883	0.547
Nutch	0.941	0.276	0.951	0.217
Wicket	0.843	0.657	0.678	0.417
Cxf	0.801	0.698	0.928	0.468
OpenJpa	0.965	0.503	0.903	0.399
Cross-Project	0.816	0.592	0.806	0.573

Evaluation

	ARC		ACDC	
	Precision	Recall	Precision	Recall
Hadoop	0.793	0.637	**0.883**	0.547
Nutch	**0.941**	0.276	**0.951**	0.217
Wicket	**0.843**	0.657	0.678	0.417
Cxf	**0.801**	0.698	**0.928**	0.468
OpenJpa	**0.965**	0.503	**0.903**	0.399
Cross-Project	**0.816**	0.592	**0.806**	0.573

Evaluation

	ARC		ACDC	
	Precision	Recall	Precision	Recall
Hadoop	0.793	0.637	0.883	0.547
Nutch	0.941	**0.276**	0.951	**0.217**
Wicket	0.843	0.657	0.678	0.417
Cxf	0.801	0.698	0.928	0.468
OpenJpa	0.965	**0.503**	0.903	**0.399**
Cross-Project	0.816	0.592	0.806	0.573

MSR 2018

Toward Predicting Architectural Significance of Implementation Issues

ARMAN SHAHBAZIAN, DAYE NAM, NENAD MEDVIDOVIC

UNIVERSITY OF SOUTHERN CALIFORNIA

Contributions

Detection

Automatic Detection of Architecturally Significant Issues

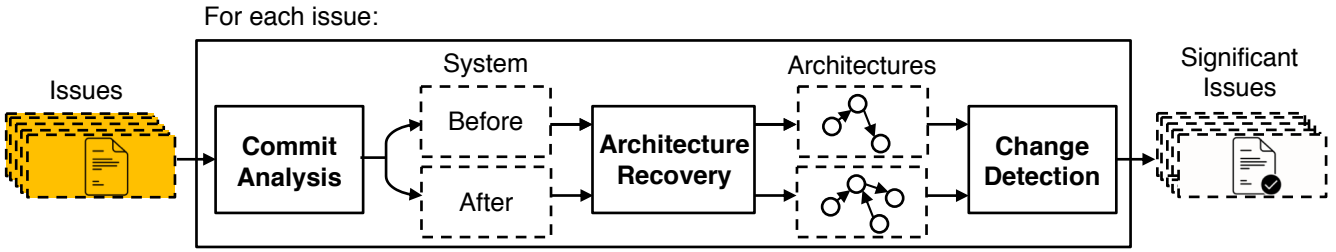
Dataset

A Dataset of 21,062 Issues Identified Across 5 Large OSSs

Classifier

A Classifier Architectural Significance of New Issue

Data Collection



Hadoop Common / HADOOP-10893

isolated classloader on the client side

Details

Type:	New Feature	Status:	CLOSED
Priority:	Major	Resolution:	Fixed
Affects Version/s:	2.4.0	Fix Version/s:	2.6.0
Component/s:	util		
Labels:	None		
Hadoop Flags:	Reviewed		

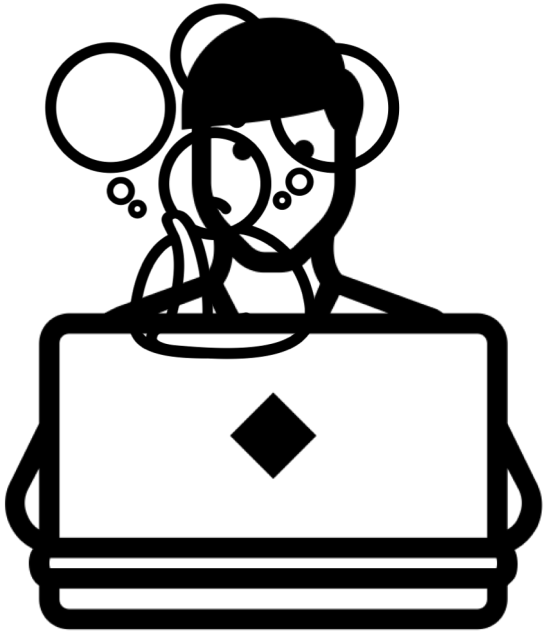
Description

We have the job classloader on the mapreduce tasks that run on the cluster. It has a benefit of being able to isolate class space for user code and avoid version clashes.

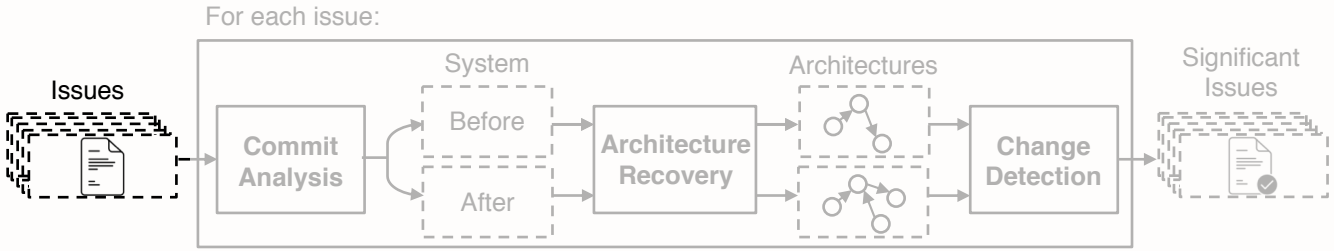
Although it occurs less often, version clashes do occur on the client JVM. It would be good to introduce an isolated classloader on the client side as well to address this. A natural point to introduce this may be through RunJar, as that's how most of hadoop jobs are run.

Future Works

Motivation



Data Collection

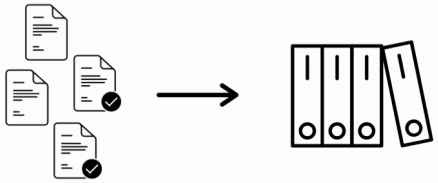


Contributions

Detection

Automatic Detection of Architecturally Significant Issues

Dataset



A Dataset of 21,062 Issues Identified Across 5 Large OSSs

Classifier

A Classifier Architectural Significance of New Issue