TagSniff

Simplified Data Debugging for Dataflow Jobs

Powered by: RHEEM

Bertty Contreras Jorge Quiané

Zoi Kaoudi Saravanan Thirumuruganathan

2019 SOCC

berlin

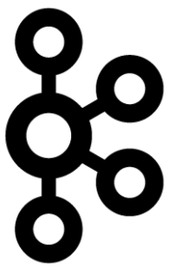
QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Many Exciting Big Data Areas!

Flink

TensorFlow

elasticsearch



kafka

Google Cloud

Amazon
Lambda

Azure Machine Learning

Debugging

...any Exciting Big Data Areas!

Flink

+ a b | e a u®

TensorFlow

elasticsearch

kafka

Google Cloud

Amazon
Lambda

Azure Machine Learning

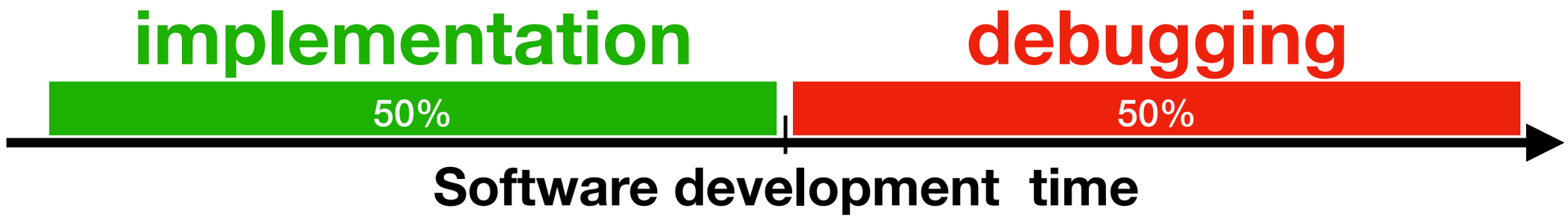
... and how is big data debugging?

… and how is big data debugging?

[The Debugging Mindset. ACM Queue, 2017.]

[Reversible debugging software “quantify the time and cost saved using reversible debuggers”. CiteSeerX, 2013]

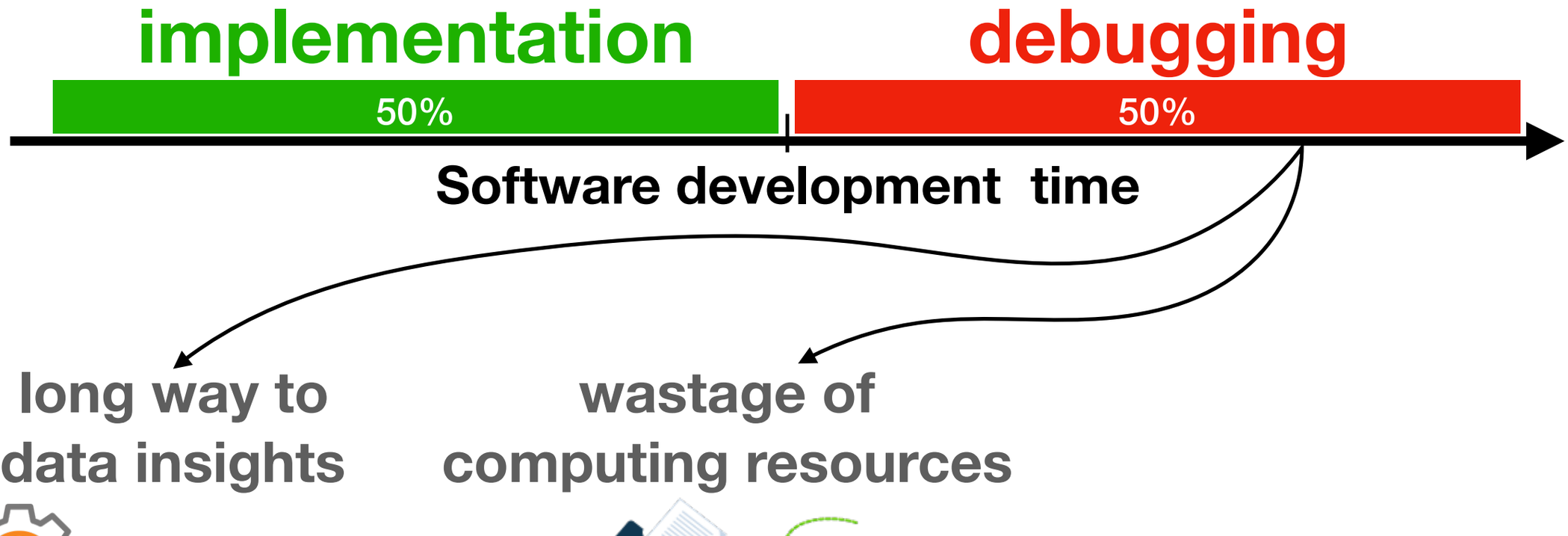
… and how is big data debugging?



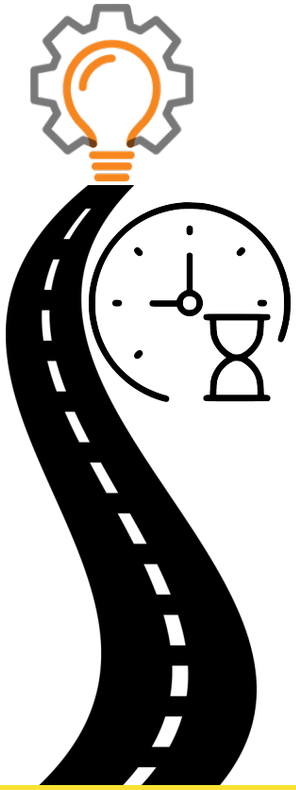
[The Debugging Mindset. ACM Queue, 2017.]

[Reversible debugging software “quantify the time and cost saved using reversible debuggers”. CiteSeerX, 2013]

… and how is big data debugging?



long way to data insights

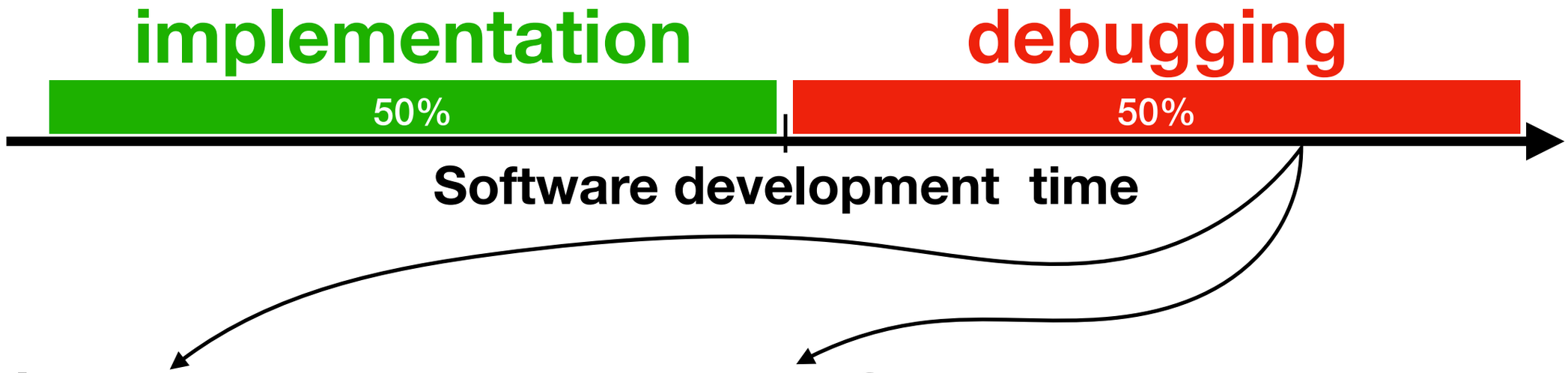


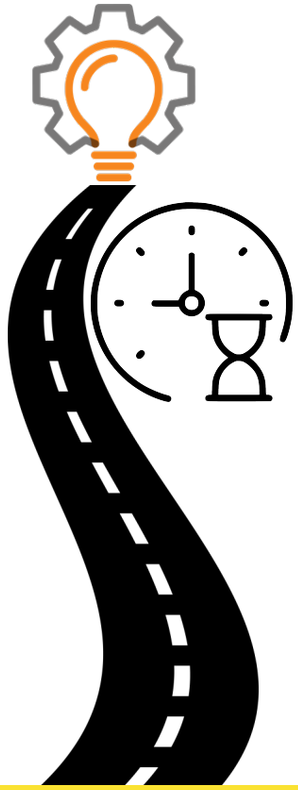
wastage of computing resources

[The Debugging Mindset. ACM Queue, 2017.]

[Reversible debugging software “quantify the time and cost saved using reversible debuggers”. CiteSeerX, 2013]

… and how is big data debugging?



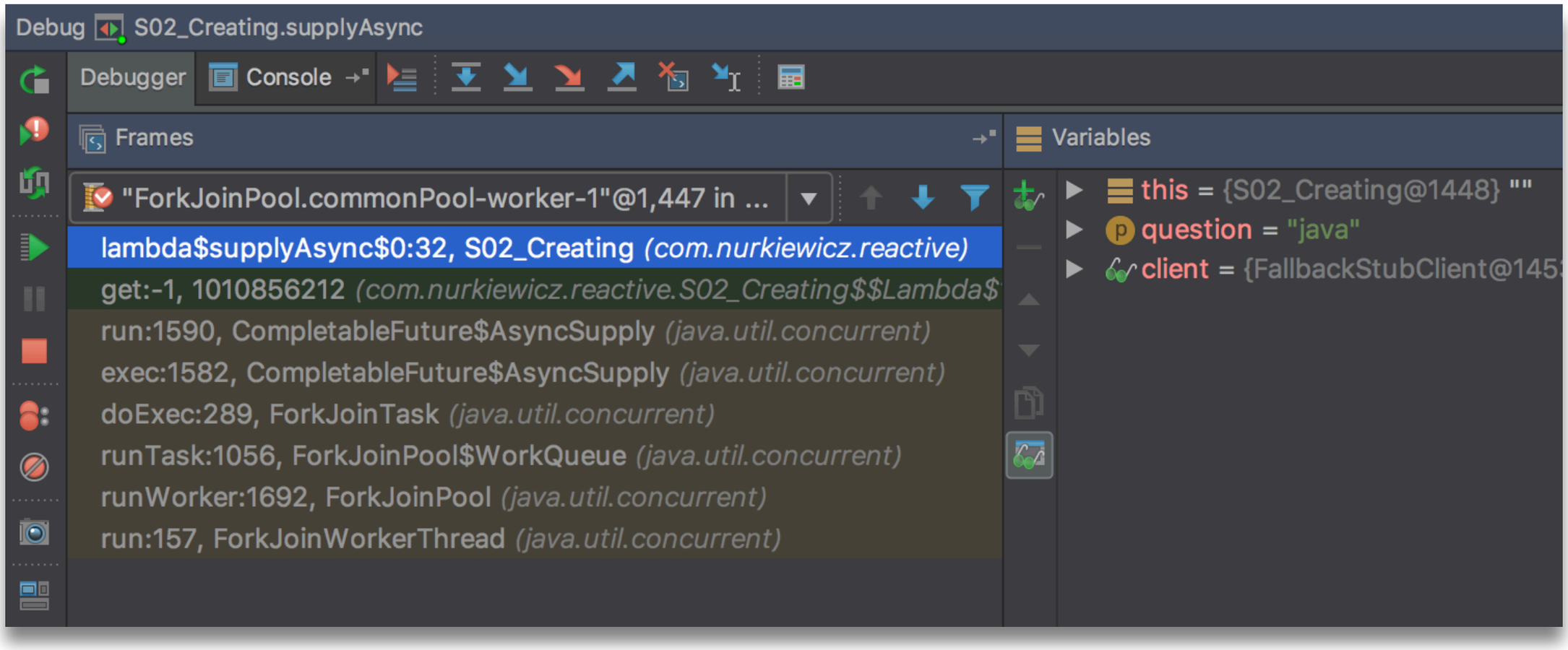


$312b/year

[The Debugging Mindset. ACM Queue, 2017.]

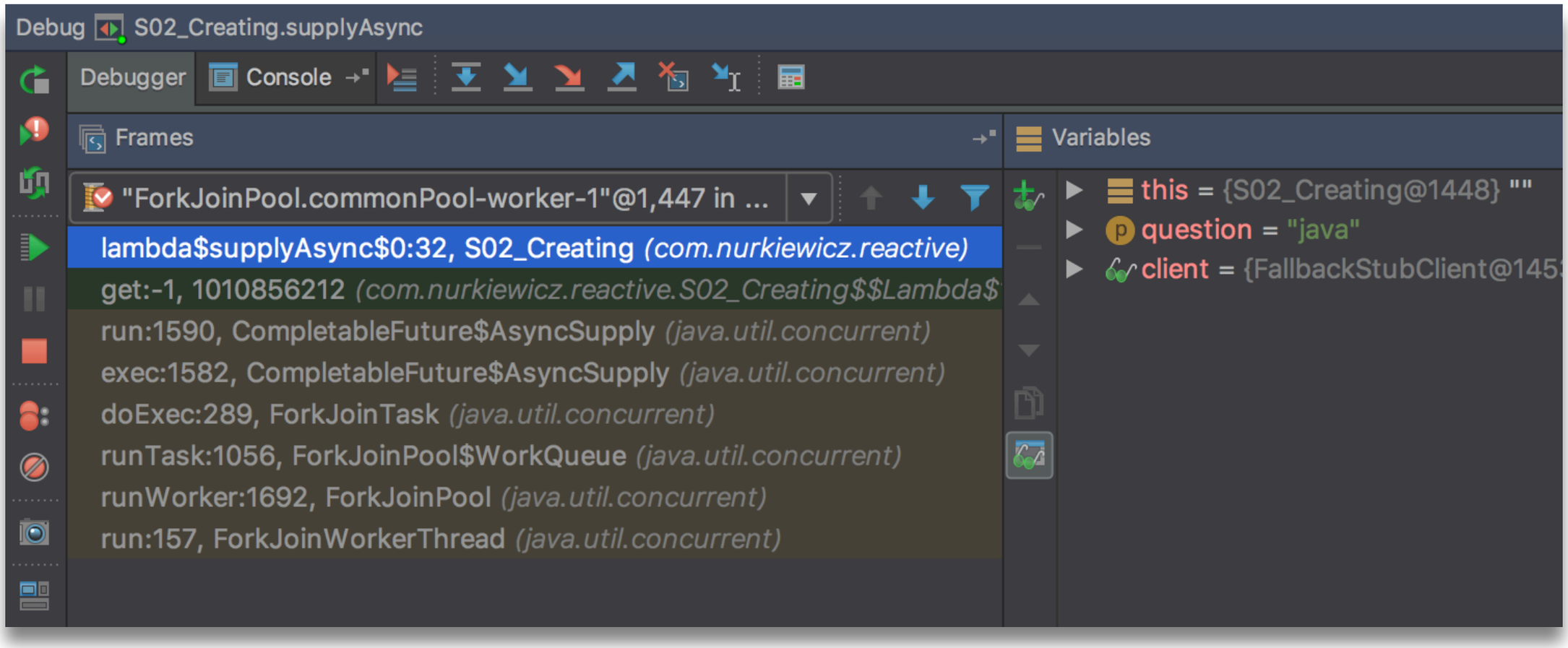
[Reversible debugging software “quantify the time and cost saved using reversible debuggers”. CiteSeerX, 2013]

Approach 1: Code Debugging



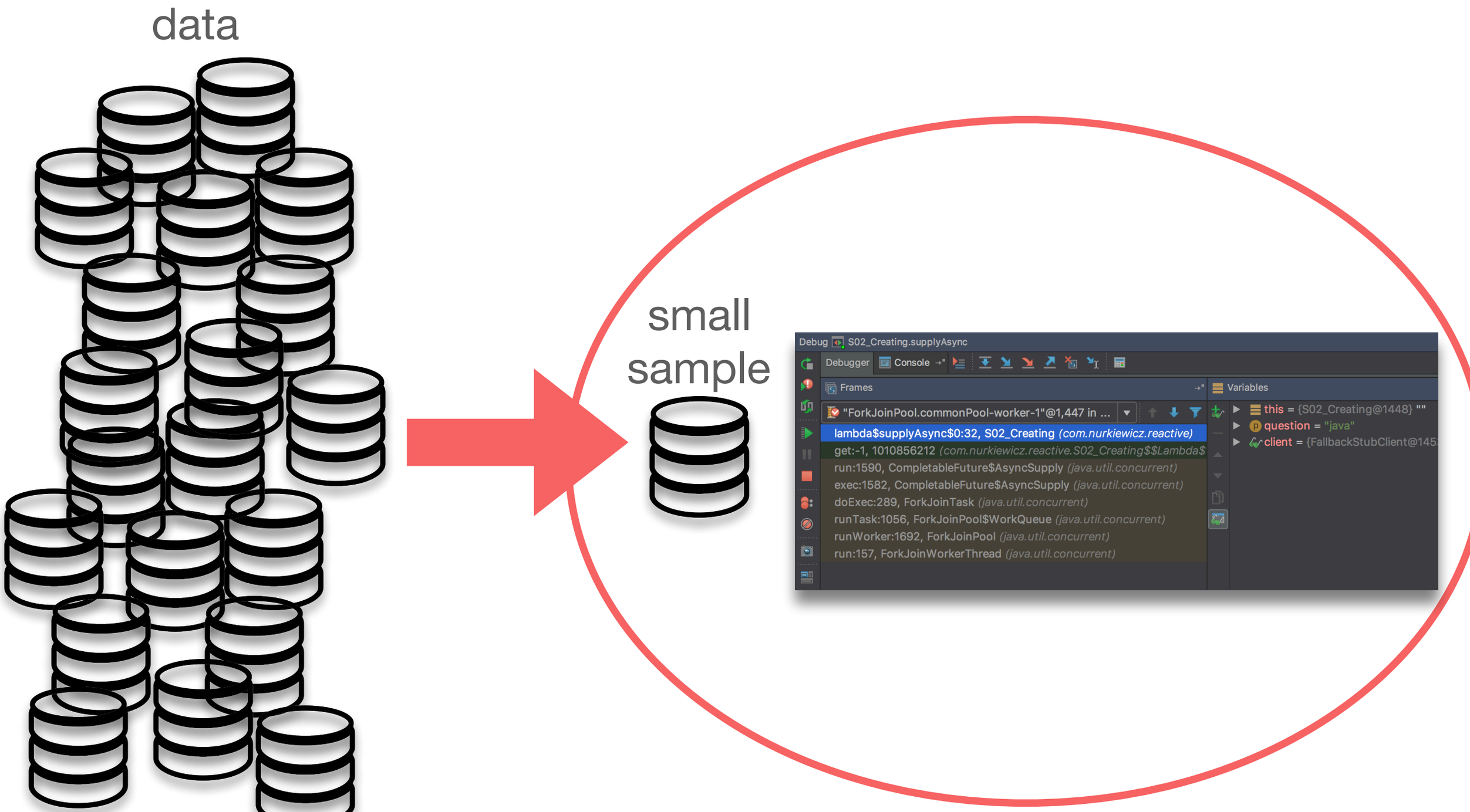
- Great for **small** data and **single** machine

Approach 1: Code Debugging

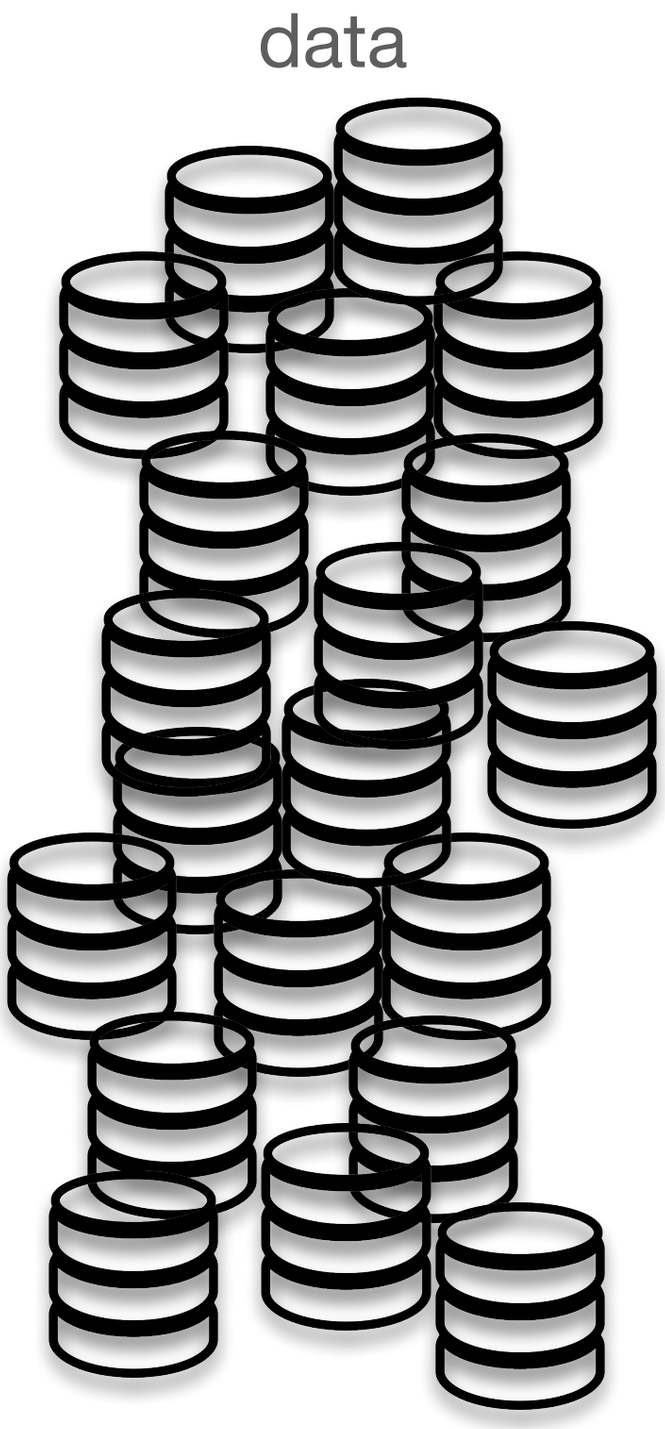


- Great for **small** data and **single** machine
- **Cannot** examine
  - line per line
  - tuple per tuple
  - machine per machine

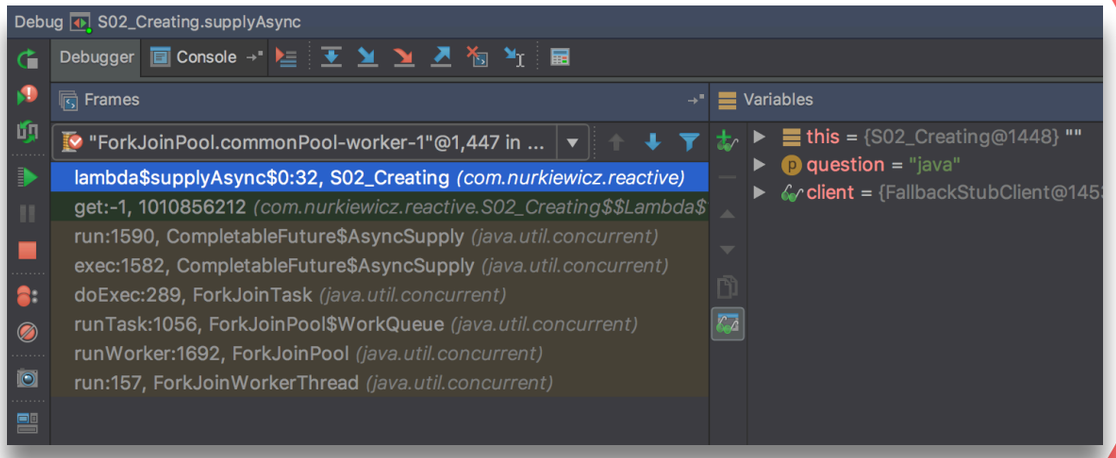
Approach 2: Sample + Code Debugging



Approach 2: Sample + Code Debugging

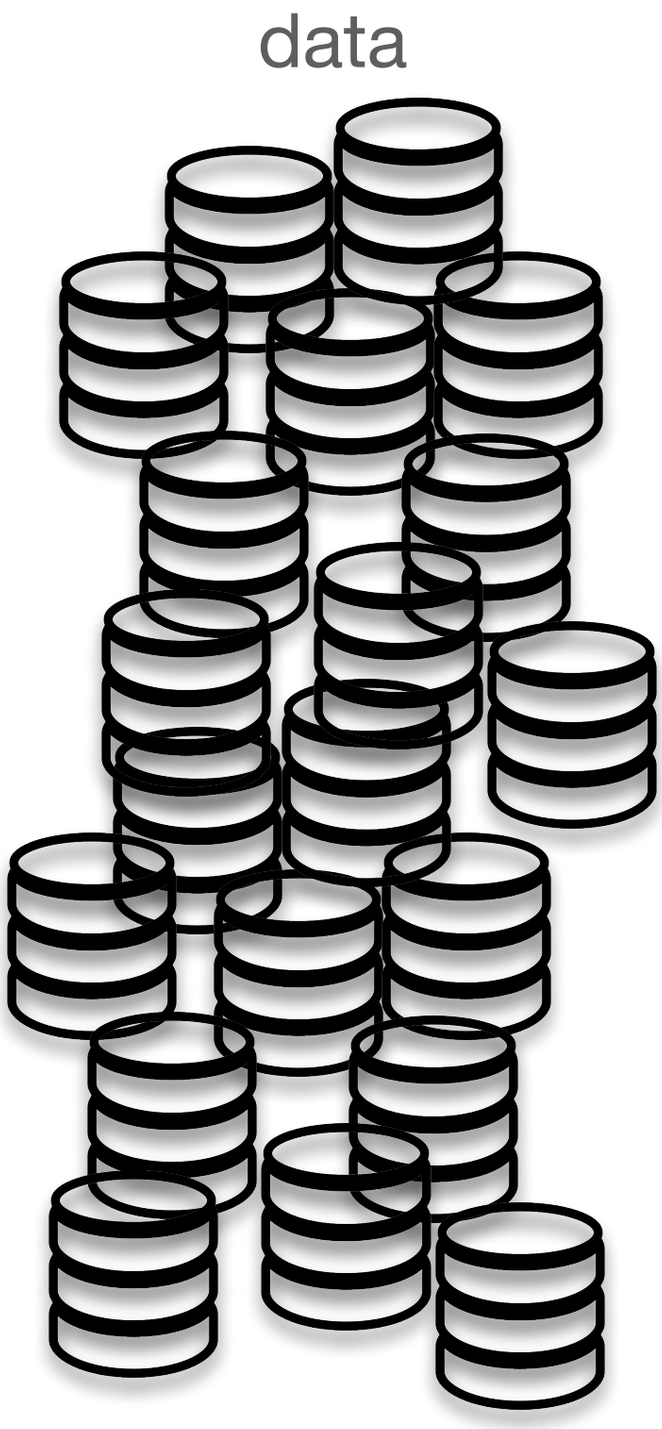


small sample

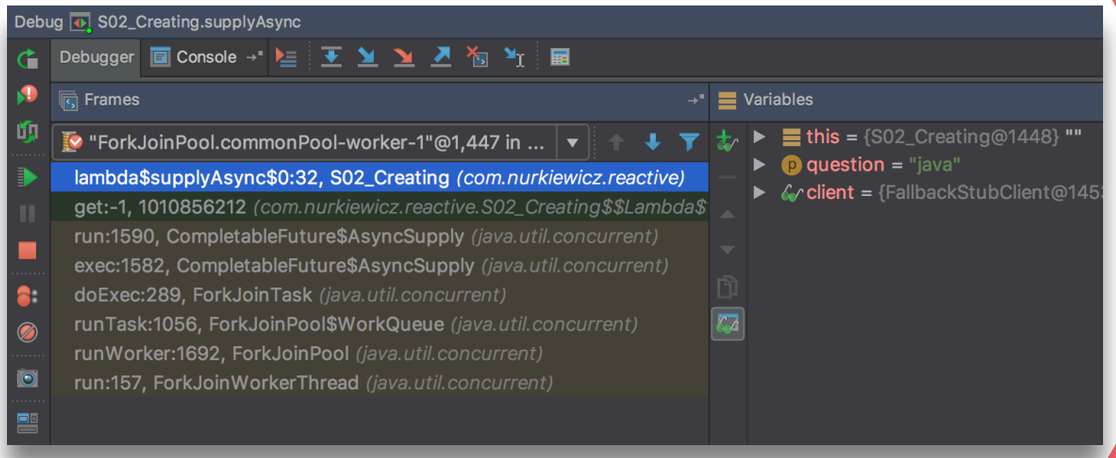


- **Hard** to spot the error!

Approach 2: Sample + ~~Code~~ Debugging Data



small sample



- **Hard** to spot the error!

Data Debugging

Identifying and removing errors stemming from the interplay between code and data

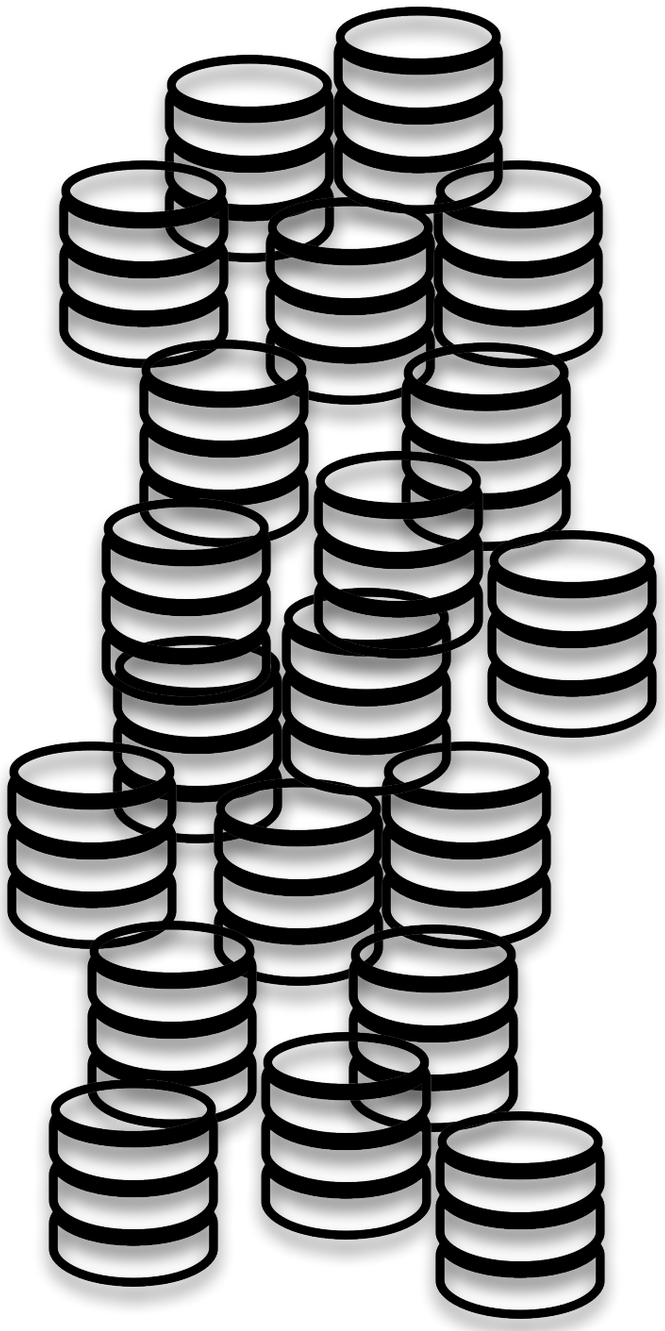
code

```
package org.alfresco.module.dbiservicesamp;

import java.io.IOException;

public class ArchiveWebscript extends AbstractWebScript {
    private Repository repository;
    private NodeService nodeService;
private FileFolderService fileFolderService;
private PermissionService permissionService;
private final static Logger logger = Logger.getLogger(ArchiveWebscript.class.getName());

public void setPermissionService(PermissionService permissionService) {
    this.permissionService = permissionService;
}

public void setFileFolderService(FileFolderService fileFolderService) {
    this.fileFolderService = fileFolderService;
}

public void setRepository(Repository repository){
    this.repository = repository;
}

public void setNodeService(NodeService nodeService) {
    this.nodeService = nodeService;
}

@Override
public void execute(WebScriptRequest req, WebScriptResponse
    logger.setLevel(Level.INFO);

    // Get the NodeRef of the node which call the webscript (document
    String nodeRefStr = req.getParameter("nodeRef");
    if (StringUtils.isBlank(nodeRefStr)) {
        throw new WebScriptException("Document nodeRef parame
    }
    final NodeRef nodeRef = new NodeRef(nodeRefStr);

    // Get the name associate to the nodeRef
    final Serializable filename = nodeService.getProperty(nodeRef, ContentMode
    if (filename == null) {
        throw new WebScriptException("Document filename is null");
    }

    // Do the work for this Web Script
    AuthenticationUtil.runAs(new AuthenticationUtil.RunAsWork<FileInfo>() {

        public FileInfo doWork() throws Exception {
            // Set the Description property to the Source Path
            String sourcePathExtracted = nodeService.getPath(nodeRef).toString()
            String sourcePathReal = sourcePathExtracted.replaceAll("\\{[^\\}]+\\
            String sourcePath = sourcePathReal.replaceFirst("/[^/]*$", "");

            String strTarget = "workspace/SpacesStore/Company Home/99_Archives";
            NodeRef archiveRef = repository.findNodeRef("path", strTarget.split(

            String message = "oldPath = " + sourcePath;
            nodeService.setProperty(nodeRef, ContentModel.PROP_DESCRIPTION, messag
```

data

Approach 3: Sample + Debugging Tips

data



repeat

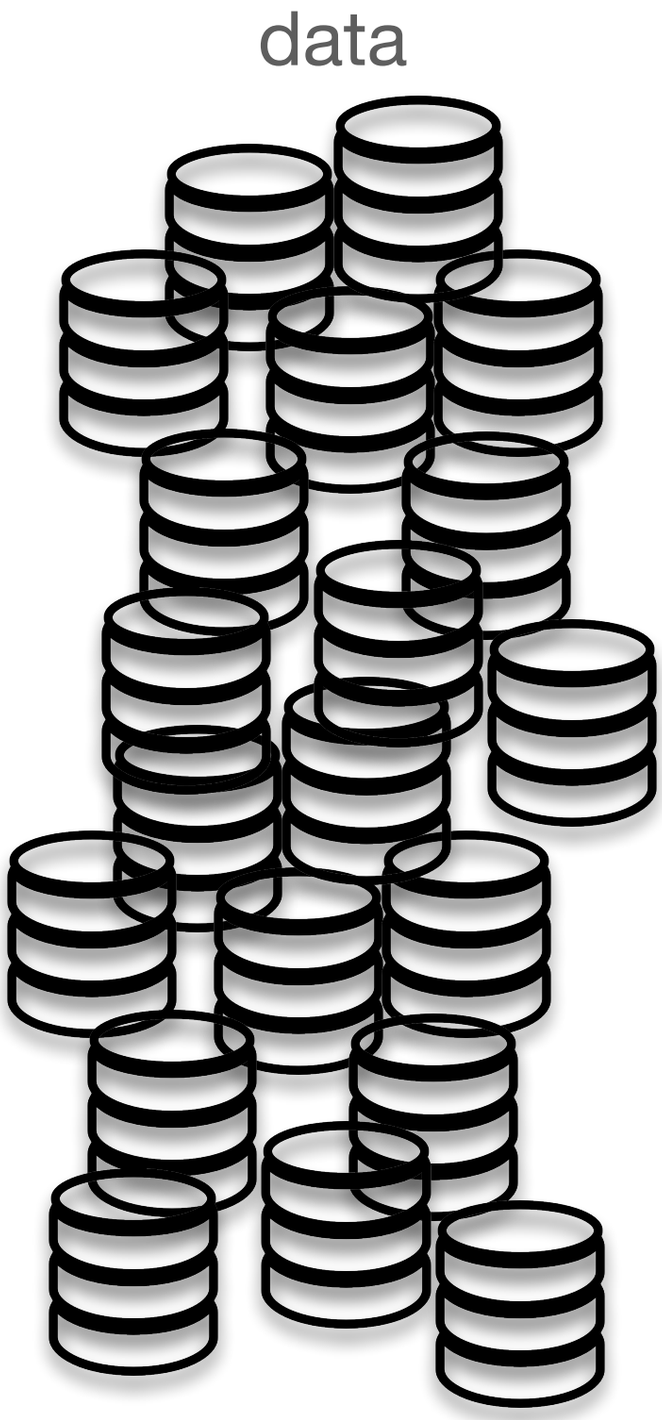
sample

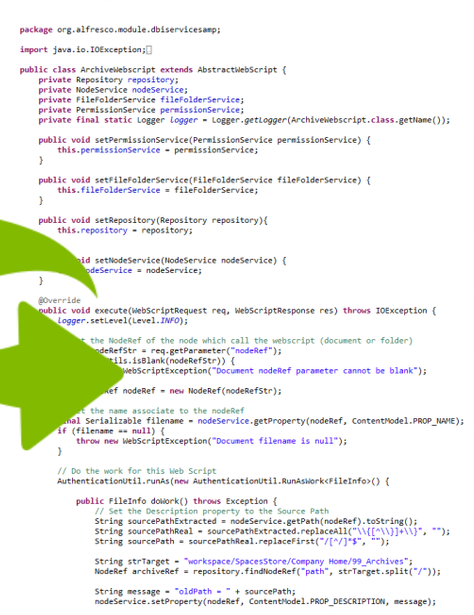
code

```
package org.alfresco.module.dbiservicesamp;

import java.io.IOException;

public class ArchiveWebscript extends AbstractWebScript {
    private Repository repository;
    private NodeService nodeService;
    private FileFolderService fileFolderService;
    private PermissionService permissionService;
    private final static Logger logger = Logger.getLogger(ArchiveWebscript.class.getName());

    public void setPermissionService(PermissionService permissionService) {
        this.permissionService = permissionService;
    }

    public void setFileFolderService(FileFolderService fileFolderService) {
        this.fileFolderService = fileFolderService;
    }

    public void setRepository(Repository repository){
        this.repository = repository;
    }

    public void setNodeService(NodeService nodeService) {
        this.nodeService = nodeService;
    }

    @Override
    public void execute(WebScriptRequest req, WebScriptResponse res) throws IOException {
        logger.setLevel(Level.INFO);

        // get the NodeRef of the node which call the webscript (document or folder)
        String nodeRefStr = req.getParameter("nodeRef");
        if (StringUtils.isBlank(nodeRefStr)) {
            throw new WebScriptException("Document nodeRef parameter cannot be blank");
        }
        NodeRef nodeRef = new NodeRef(nodeRefStr);

        // get the name associate to the nodeRef
        final Serializable filename = nodeService.getProperty(nodeRef, ContentModel.PROP_NAME);
        if (filename == null) {
            throw new WebScriptException("Document filename is null");
        }

        // Do the work for this Web Script
        AuthenticationUtil.runAs(new AuthenticationUtil.RunAsWork<FileInfo>() {

            public FileInfo doWork() throws Exception {
                // Set the Description property to the Source Path
                String sourcePathExtracted = nodeService.getPath(nodeRef).toString();
                String sourcePathReal = sourcePathExtracted.replaceAll("\\{[^\\}]+\\}", "");
                String sourcePath = sourcePathReal.replaceFirst("/[^/]*$", "");

                String strTarget = "workspace/SpacesStore/Company Home/99_Archives";
                NodeRef archiveRef = repository.findNodeRef("path", strTarget.split("/"));

                String message = "oldPath = " + sourcePath;
                nodeService.setProperty(nodeRef, ContentModel.PROP_DESCRIPTION, message);
```

instrumentation

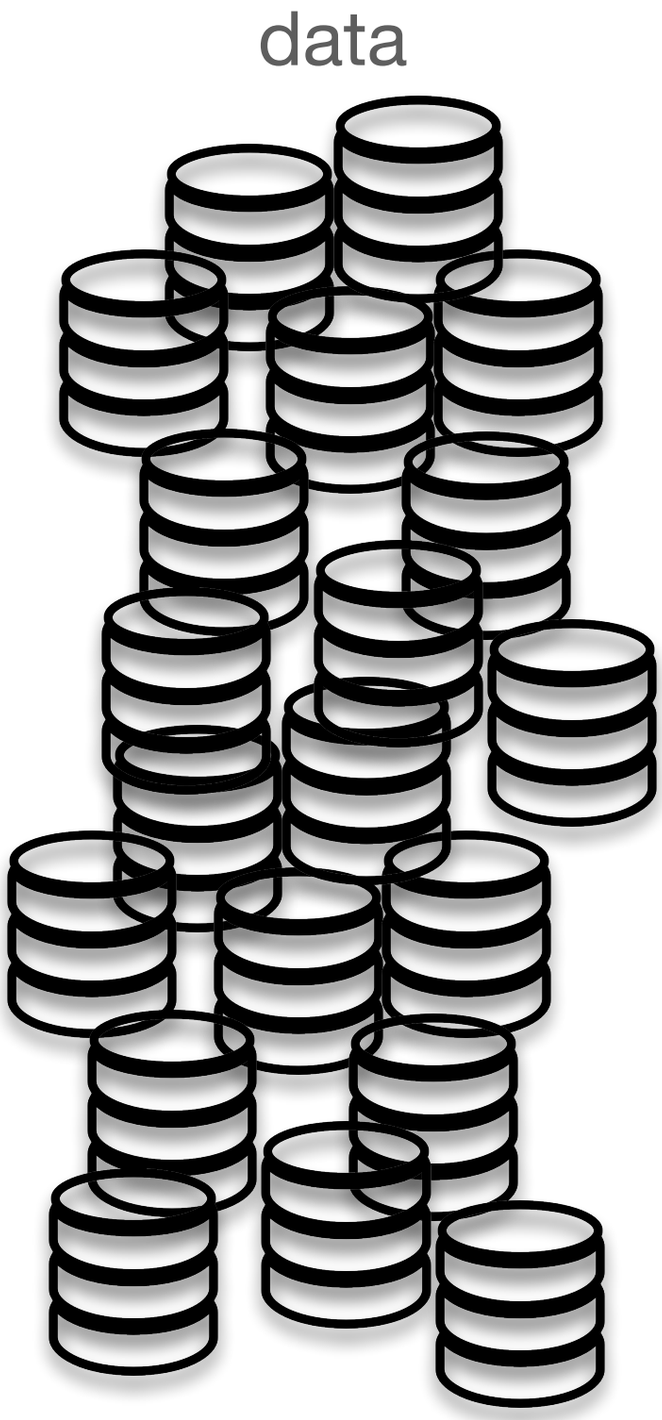
Approach 3: Sample + Debugging Tips





- **Hard** to spot the error
- **Time-consuming** task

Approach 3: Sample + Debugging Tips



- **Hard** to spot the error
- **Time-consuming** task
- Doomed to **fail**!

Need for Data Debugging Systems

Debugging mode	Task
Online	crash culprit
	pause
	alert
	replay
Post-hoc	trace
	profile
	assert

[An empirical study on quality issues of production big data platform. ICSE, 2015]

[Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows. PVLDB, 2011]

Need for Data Debugging Systems

Debugging mode	Task
Online	crash culprit
	pause
	alert
	replay
Post-hoc	trace
	profile
	assert

[BigDebug: Debugging Primitives for Interactive Big Data Processing in Spark. ICSE, 2016]

[Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows. PVLDB, 2011]

[Scalable Lineage Capture for Debugging DISC Analytics. SoCC, 2013]

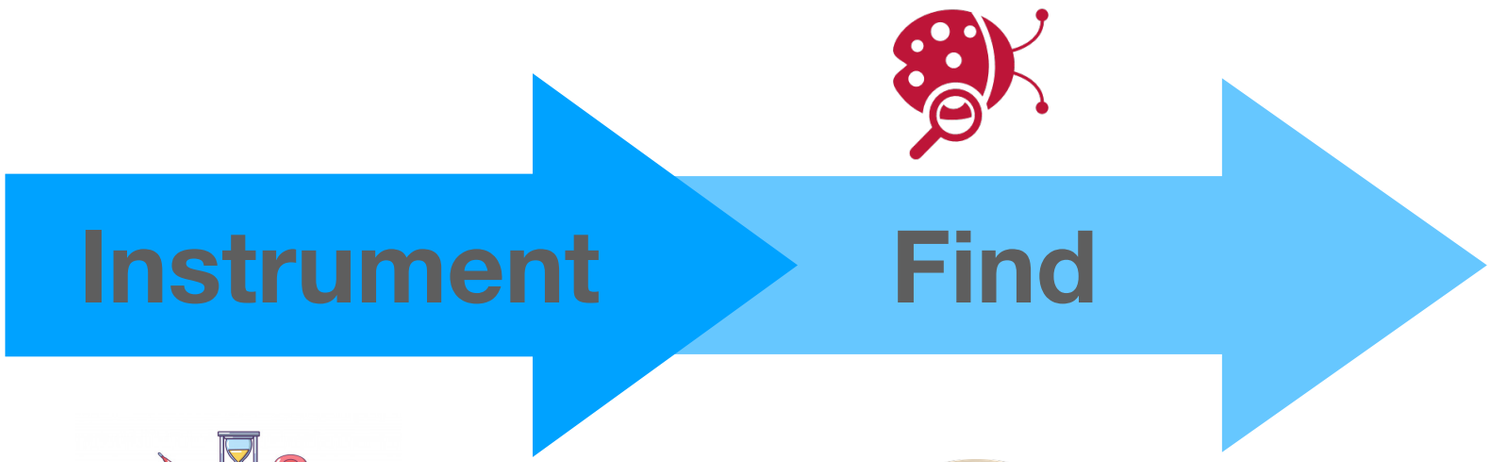
[Arthur: Rich Post-Facto Debugging for Production Analytics Applications. TR-UCBerkeley, 2013]

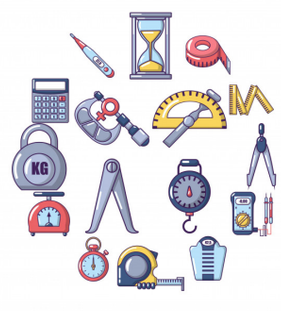
[An empirical study on quality issues of production big data platform. ICSE, 2015]

[Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows. PVLDB, 2011]

Commonalities in Data Debugging

Commonalities in Data Debugging

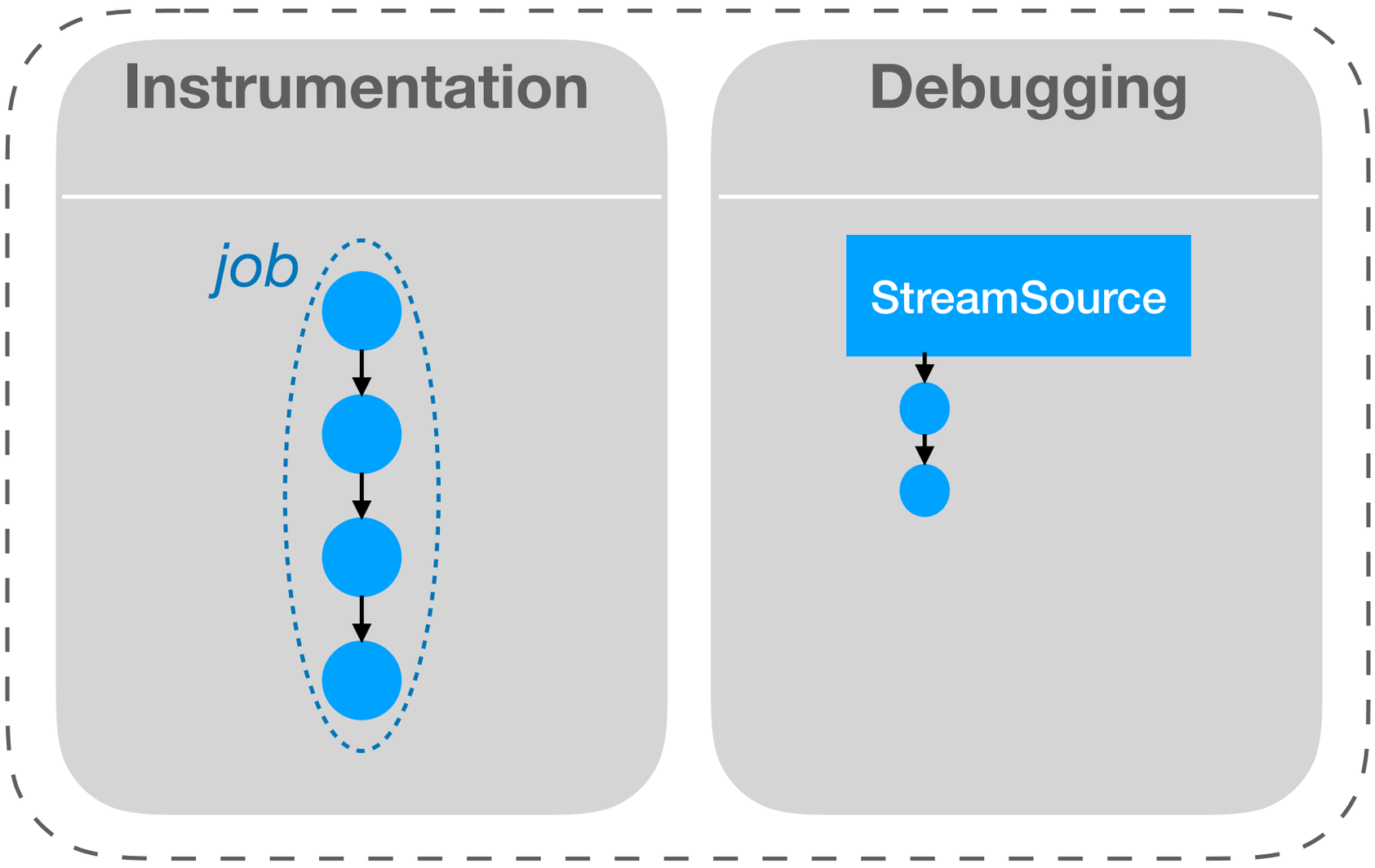




Commonalities in Data Debugging

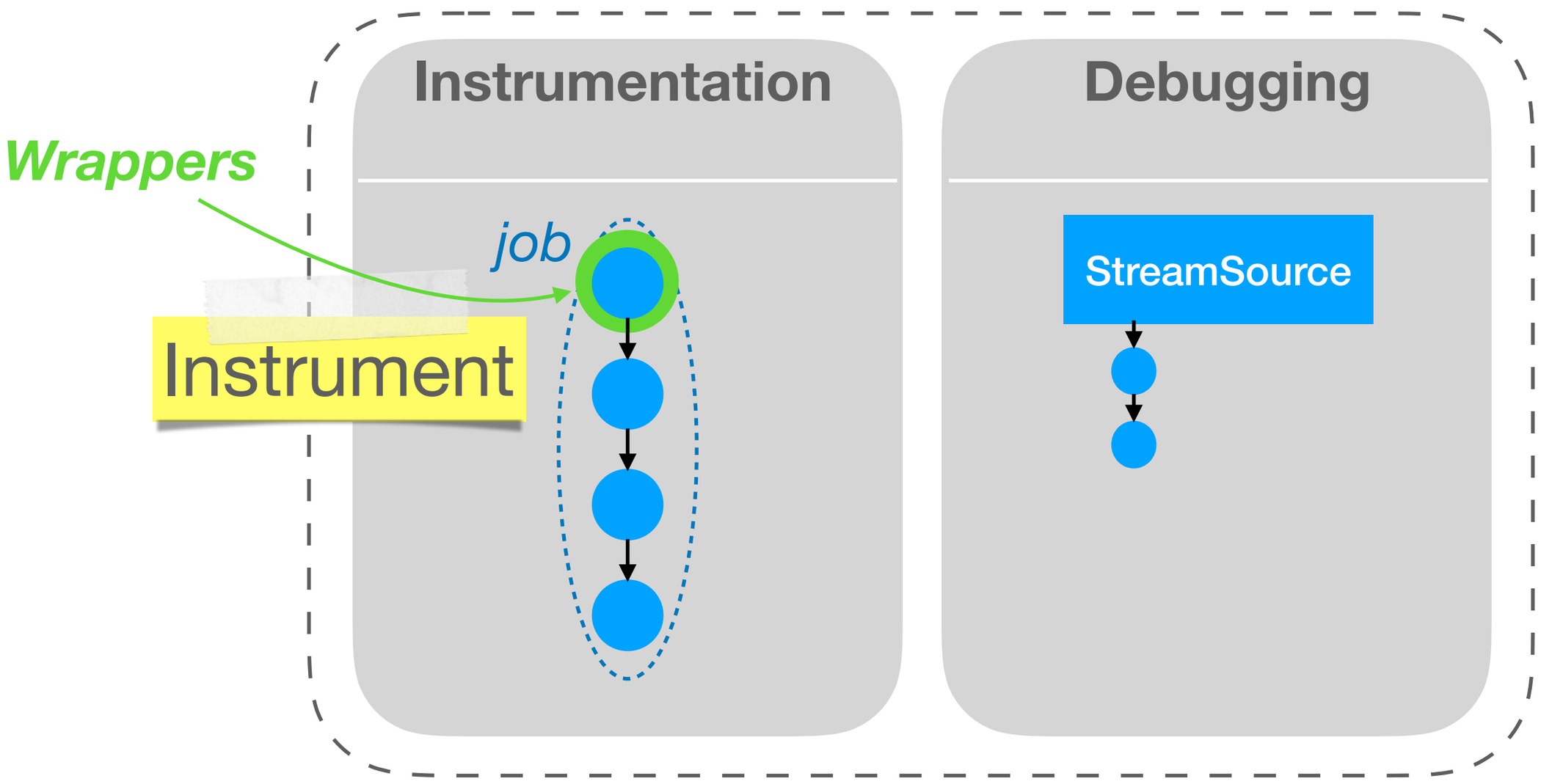
Our Idea


Instrument
Find
Act


Instrumentation
Debugging
job
StreamSource

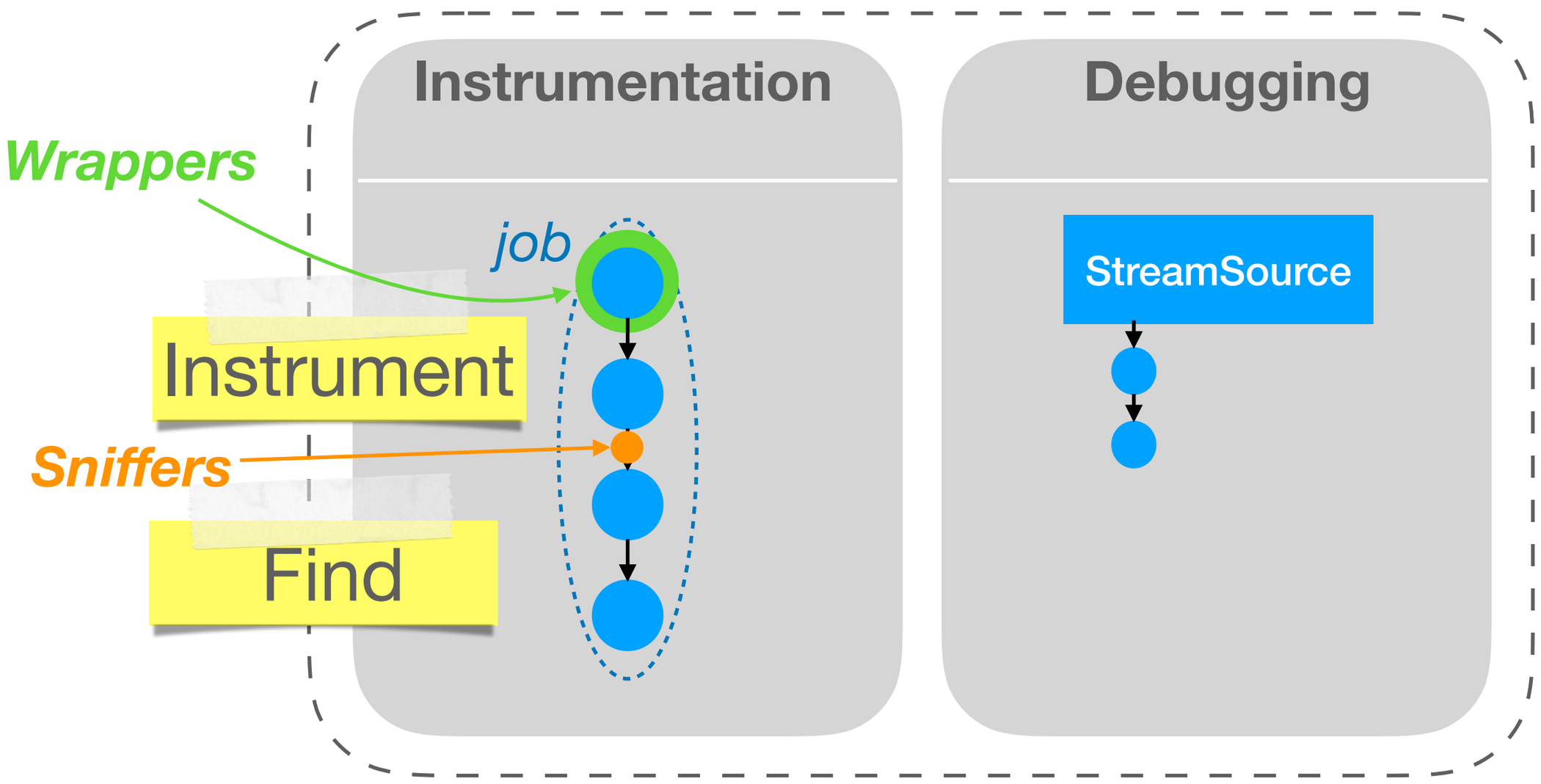
Our Idea


Instrument
Find
Act


Instrumentation
Debugging
Wrappers
job
Instrument
StreamSource

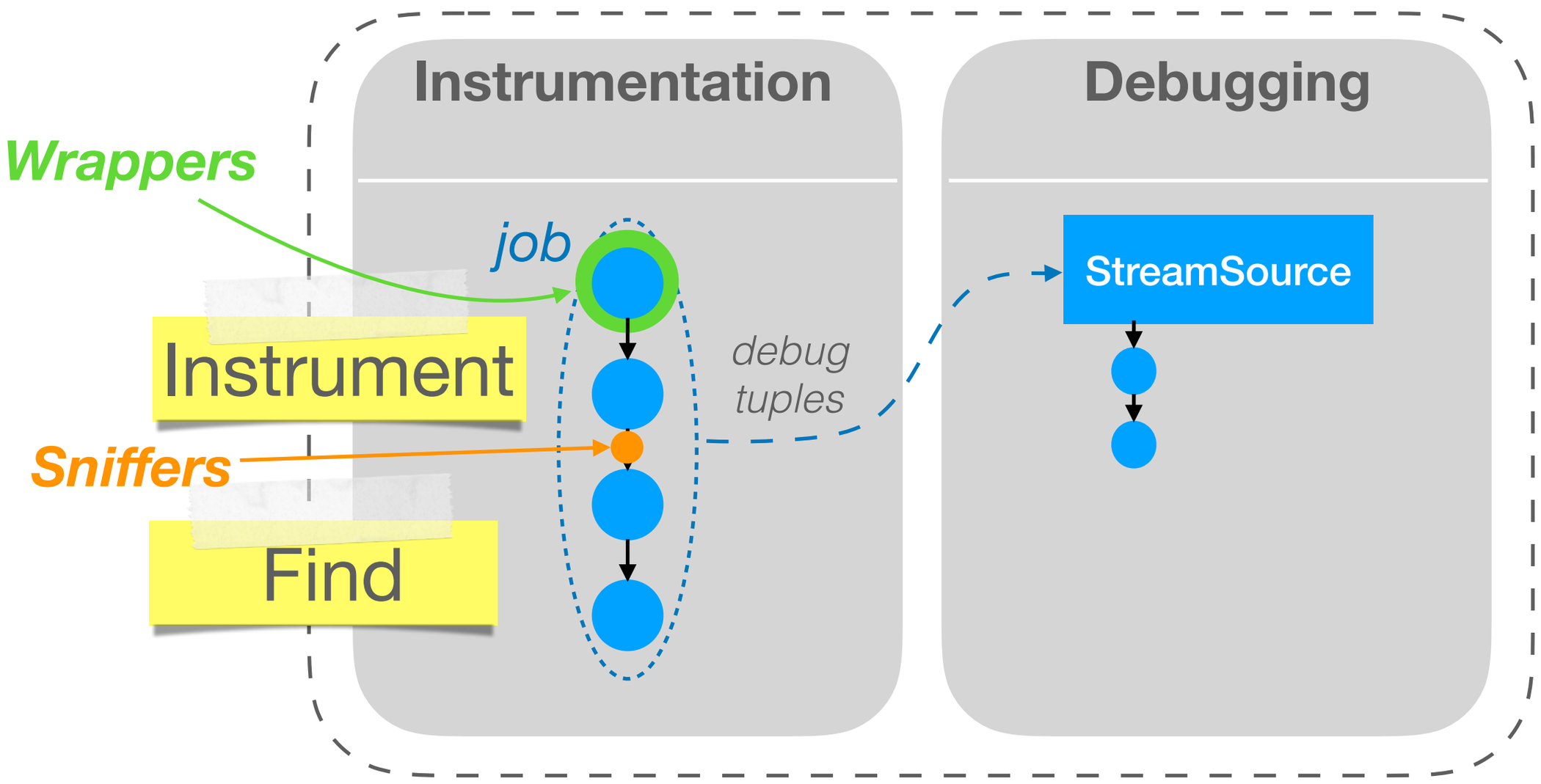
Our Idea


Instrument
Find
Act


Instrumentation
Debugging
Wrappers
job
Instrument
Sniffers
Find
StreamSource

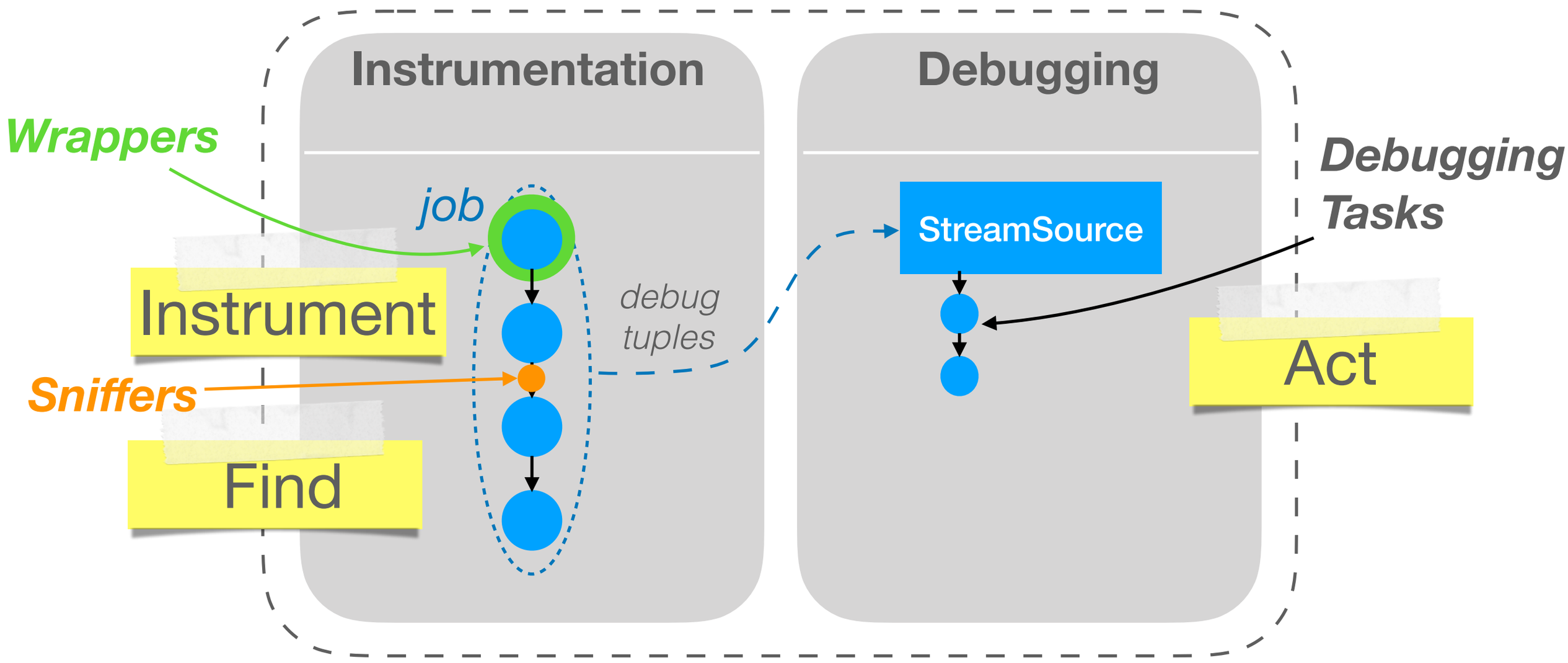
Our Idea


Instrument
Find
Act


Instrumentation
Debugging
Wrappers
job
Instrument
Sniffers
Find
debug
tuples
StreamSource

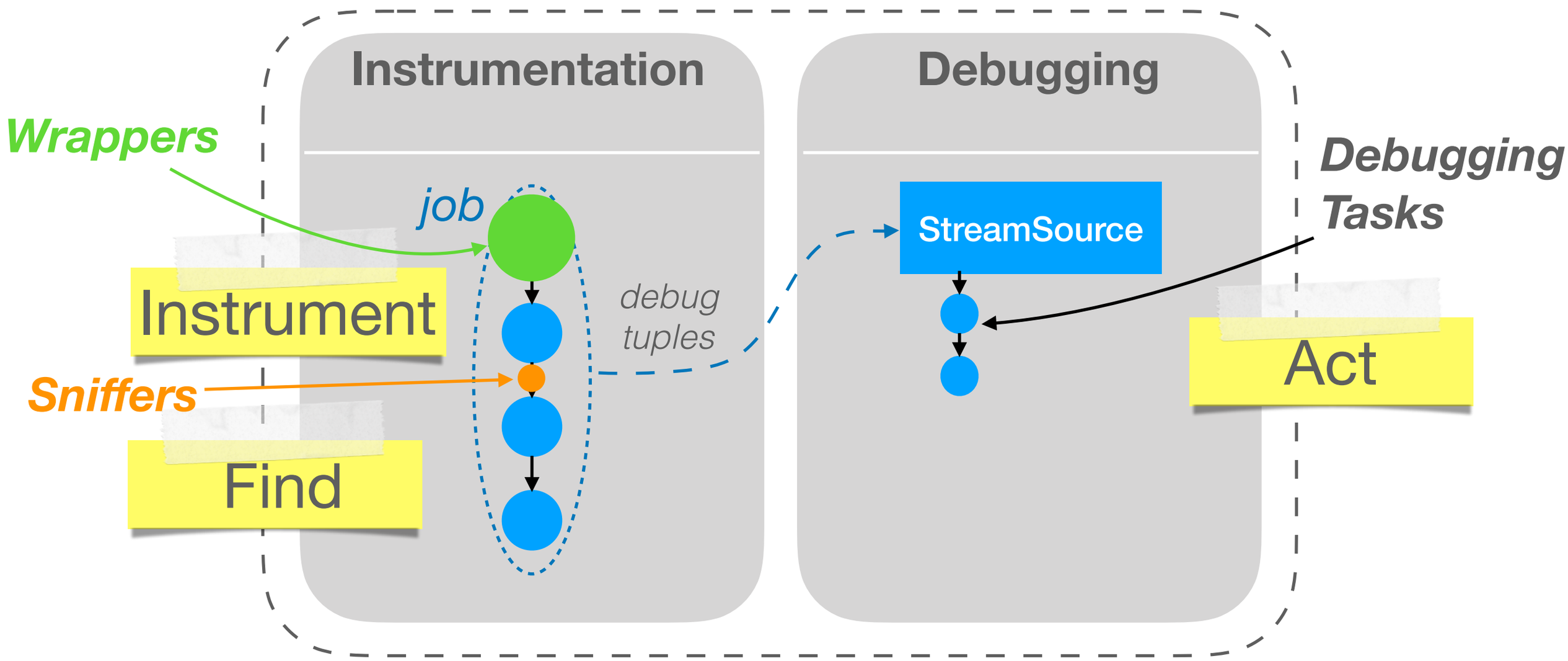
Our Idea


Instrument
Find
Act


Instrumentation
Debugging
Wrappers
job
Instrument
Sniffers
Find
debug
tuples
StreamSource
Debugging
Tasks
Act

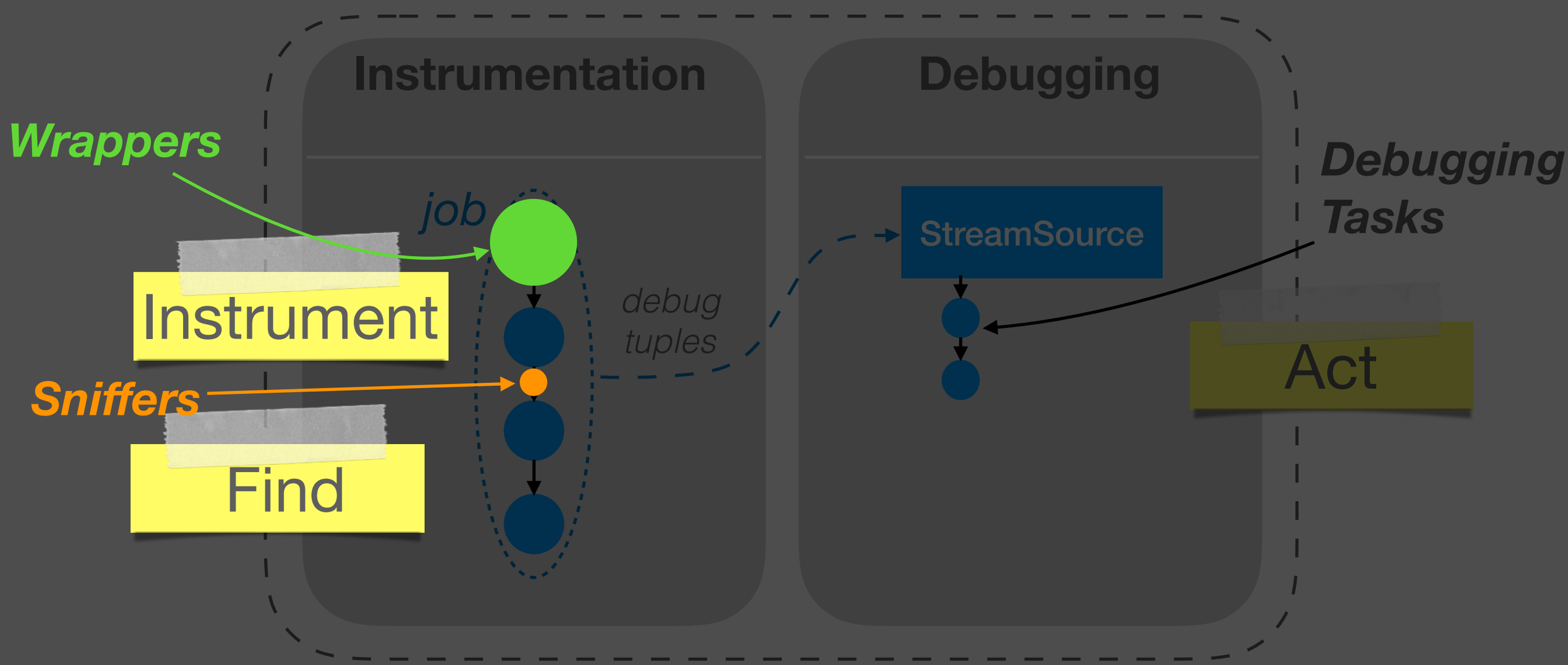
Our Idea


Instrument
Find
Act


Instrumentation
Debugging
Wrappers
job
Instrument
Sniffers
Find
debug
tuples
StreamSource
Debugging
Tasks
Act

Our Idea


Instrument
Find
Act


Instrumentation
Debugging
Wrappers
job
Instrument
Sniffers
Find
debug
tuples
StreamSource
Debugging
Tasks
Act

TagSniff Model

Desiderata for Data Debugging

(1) Support for common debugging tasks:

Online — crash culprit, pause, alert

Post-hoc — replay, trace, profile, assert

[An empirical study on quality issues of production big data platform. ICSE, 2015]

[Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows. PVLDB, 2011]

Desiderata for Data Debugging

(1) Support for common debugging tasks:

Online — crash culprit, pause, alert

Post-hoc — replay, trace, profile, assert

(2) Concise: simple to code

[An empirical study on quality issues of production big data platform. ICSE, 2015]

[Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows. PVLDB, 2011]

Desiderata for Data Debugging

(1) Support for common debugging tasks:
Online — crash culprit, pause, alert
Post-hoc — replay, trace, profile, assert

(2) Concise: simple to code

(3) Flexible:
ad-hoc debugging scenarios and monitoring

[An empirical study on quality issues of production big data platform. ICSE, 2015]

[Inspector Gadget: A Framework for Custom Monitoring and Debugging of Distributed Dataflows. PVLDB, 2011]

Primitives

tag(f:tuple => tuple): annotates tuples

Instrument

Primitives

tag(f:tuple => tuple): annotates tuples

Instrument

Code snippet:

```
tag(t => if (t.contains(null)) t.add_tag("pause"))
```

Primitives

tag(f:tuple => tuple): annotates tuples

Code snippet:

Instrument

```
tag(t => if (t.contains(null)) t.add_tag("pause"))
```

<tag1|tag2|…, <att1, att2, … >>

annotations tuple

Primitives

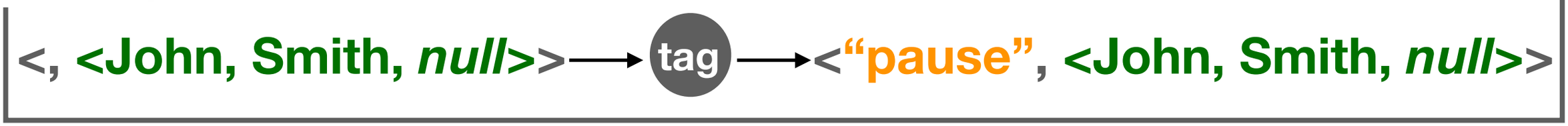
tag(f:tuple => tuple): annotates tuples

Code snippet:

Instrument

```
tag(t => if (t.contains(null)) t.add_tag("pause"))
```

Example:



Primitives

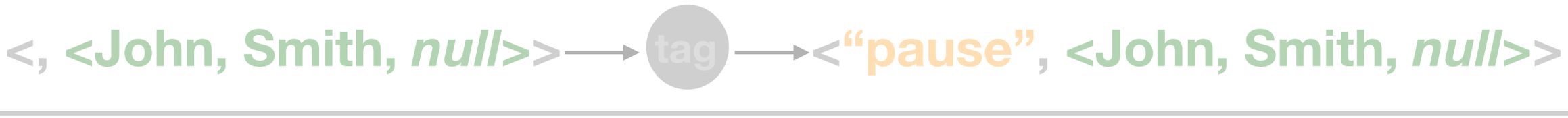
tag(f:tuple => tuple): annotates tuples

Instrument

Code snippet:

```
tag(t => if (t.contains(null)) t.add_tag("pause"))
```

Example:



sniff(f:tuple => boolean): identify tuples for reacting

Find

Primitives

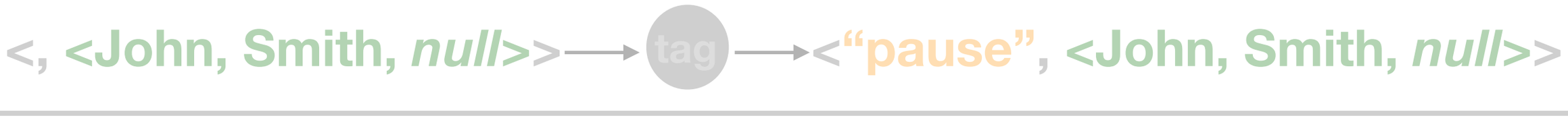
tag(f:tuple => tuple): annotates tuples

Instrument

Code snippet:

```
tag(t => if (t.contains(null)) t.add_tag("pause"))
```

Example:



sniff(f:tuple => boolean): identify tuples for reacting

Find

Code snippet:

```
sniff(t => return t.has_tag("pause"))
```

Primitives

tag(f:tuple => tuple): annotates tuples

Instrument

Code snippet:

```
tag(t => if (t.contains(null)) t.add_tag("pause"))
```

Example:

<, <John, Smith, *null*>> → tag → <“pause”, <John, Smith, *null*>>

sniff(f:tuple => boolean): identify tuples for reacting

Find

Code snippet:

```
sniff(t => return t.has_tag("pause"))
```

Example:

<“pause”, <John, Smith, *null*>> → sniff → true

TagSniff Example

tag(f:tuple => tuple): annotates tuples

Instrument

sniff(f:tuple => boolean): identify tuples for debugging

Find

WordCount

```
1   val ln = new RDDbug(spark.textFile(file))
2     .setTag(t => if (line_number(t) % 10000 == 0)
          t.add_tag("pause"))
3   val tw = ln.flatMap(l => l.split(" "))
4   val wc = tw.map(word => (word, 1))
5     .setSniff(t => return t.has_tag("pause"))
6   val wct = wc.reduceByKey(_ + _)
```

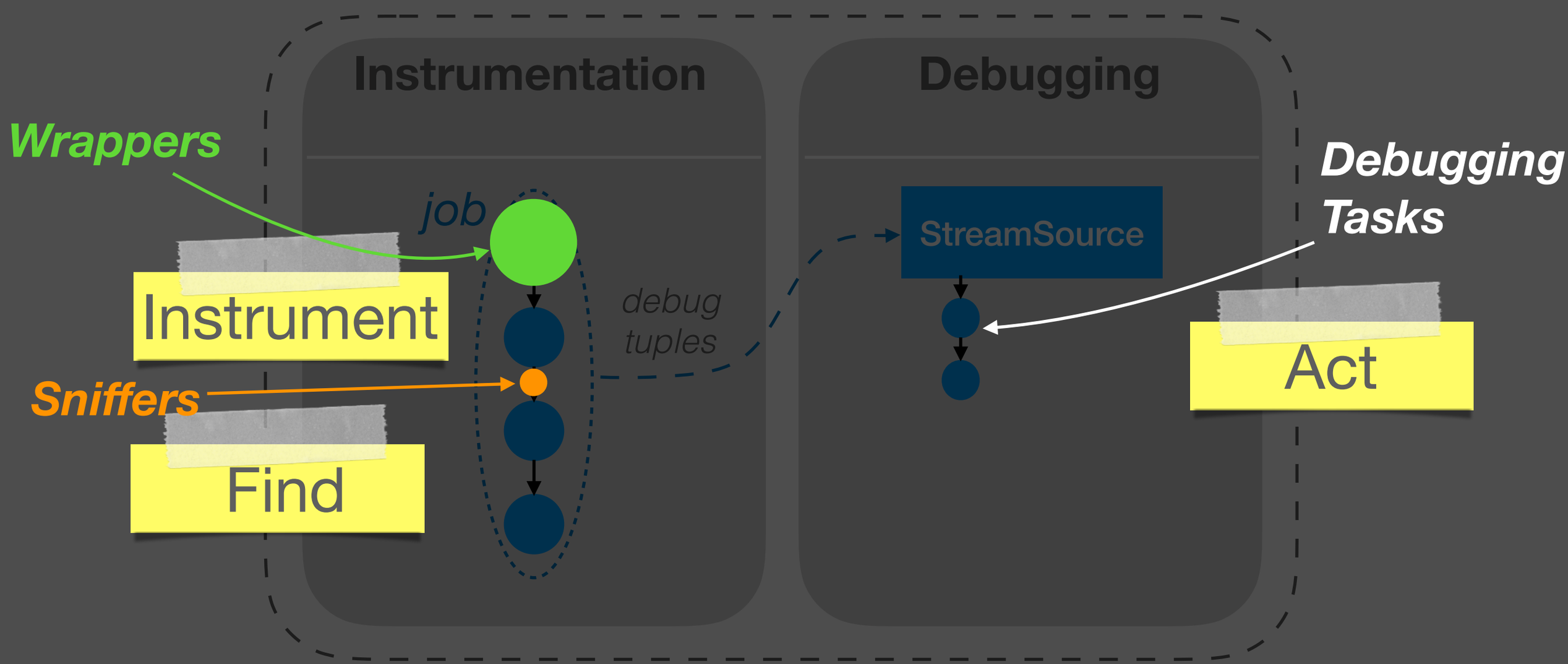
TagSniff Logging Example

```
1   tag(t => if (t.contains(null)) {
2           id = Generator.generate_id(t)
3           t.add_tag("id-"+id)
4           t.add_tag("log")})
5   sniff(t => return t.has_tag("log"))
```

Code snippet: logging all null values

Our Idea


Instrument
Find
Act

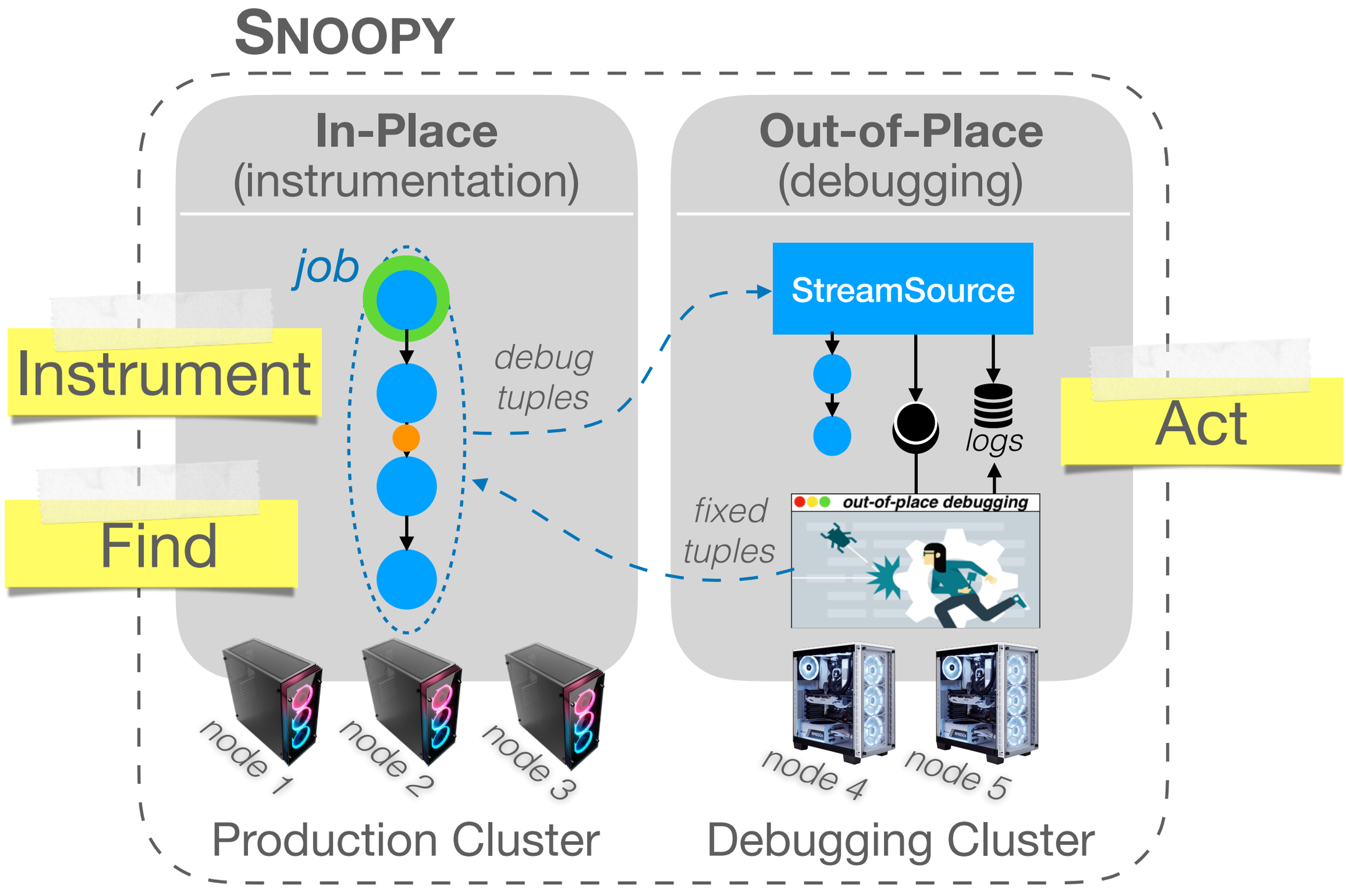

Instrumentation
Debugging
Wrappers
job
Instrument
debug
tuples
Sniffers
Find
StreamSource
Debugging
Tasks
Act

SNOOPY

— A TagSniff Instantiation for Spark —

RHEEM

Architecture

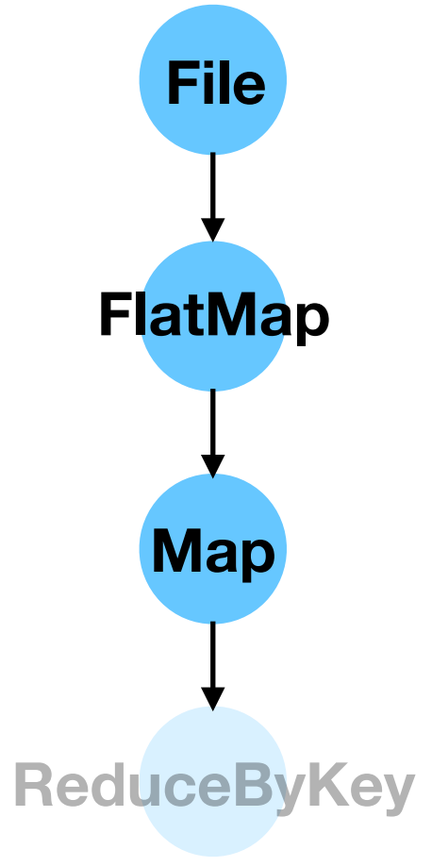


Tagging Tuples

Code snippet: pausing every 10k tuples

```
1   val ln = new RDDbug(spark.textFile(file))
2     .setTag(t => if (line_number(t) % 10000 == 0)
                t.add_tag("pause"))
3   val tw = ln.flatMap(l => l.split(" "))
4   val wc = tw.map(word => (word, 1))
5     .setSniff(t => return t.has_tag("pause"))
```

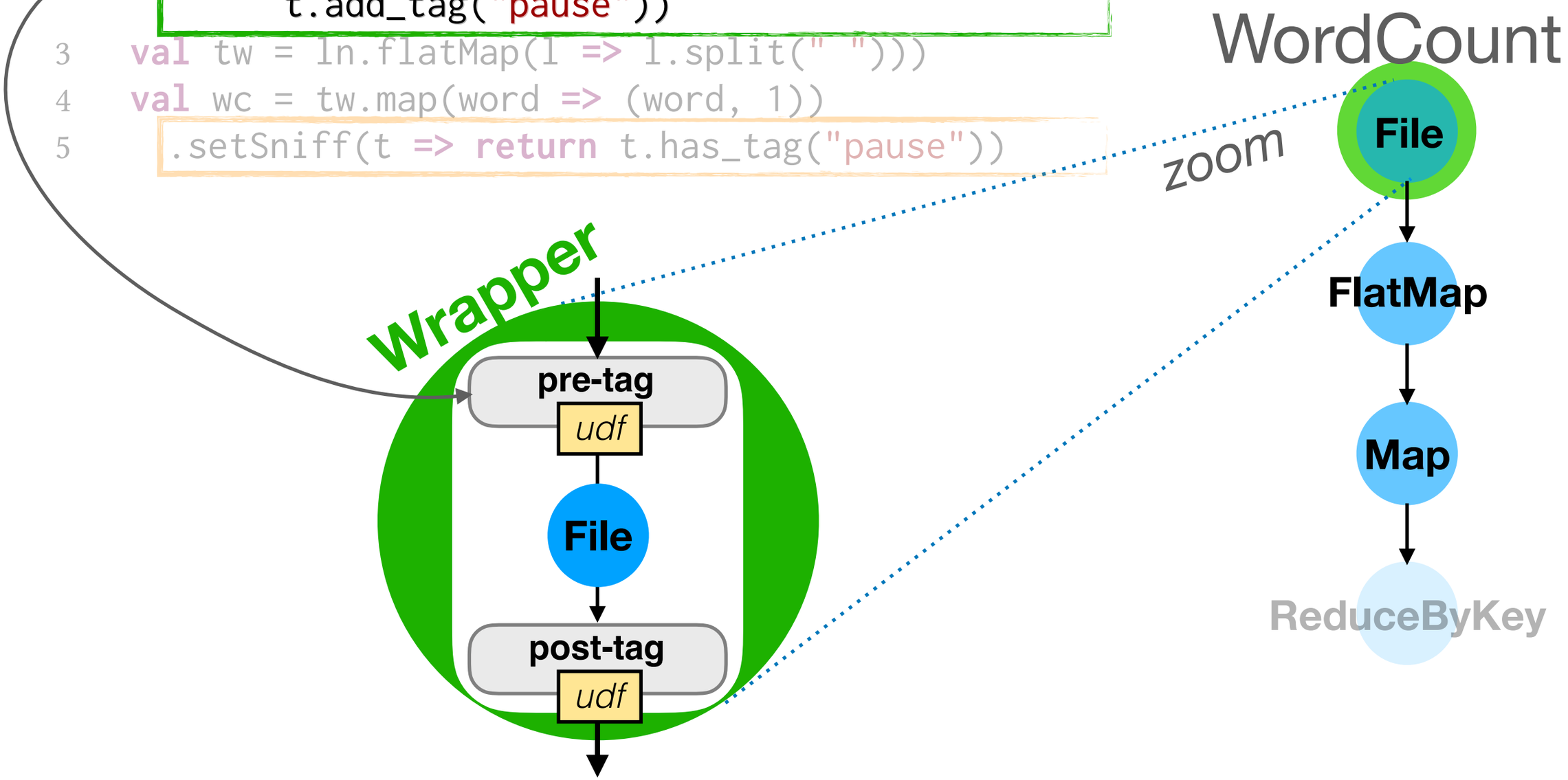
WordCount



Tagging Tuples

Code snippet: pausing every 10k tuples

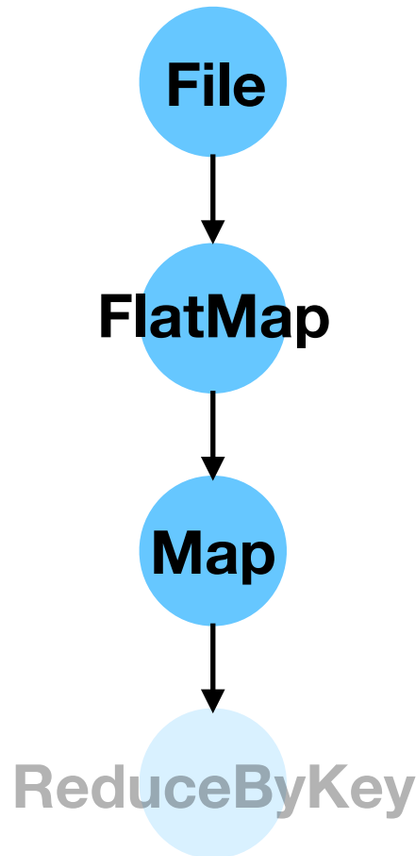
```
1   val ln = new RDDbug(spark.textFile(file))
2     .setTag(t => if (line_number(t) % 10000 == 0)
          t.add_tag("pause"))
3   val tw = ln.flatMap(l => l.split(" "))
4   val wc = tw.map(word => (word, 1))
5     .setSniff(t => return t.has_tag("pause"))
```



Sniffing Tuples

Code snippet: pausing every 10k tuples

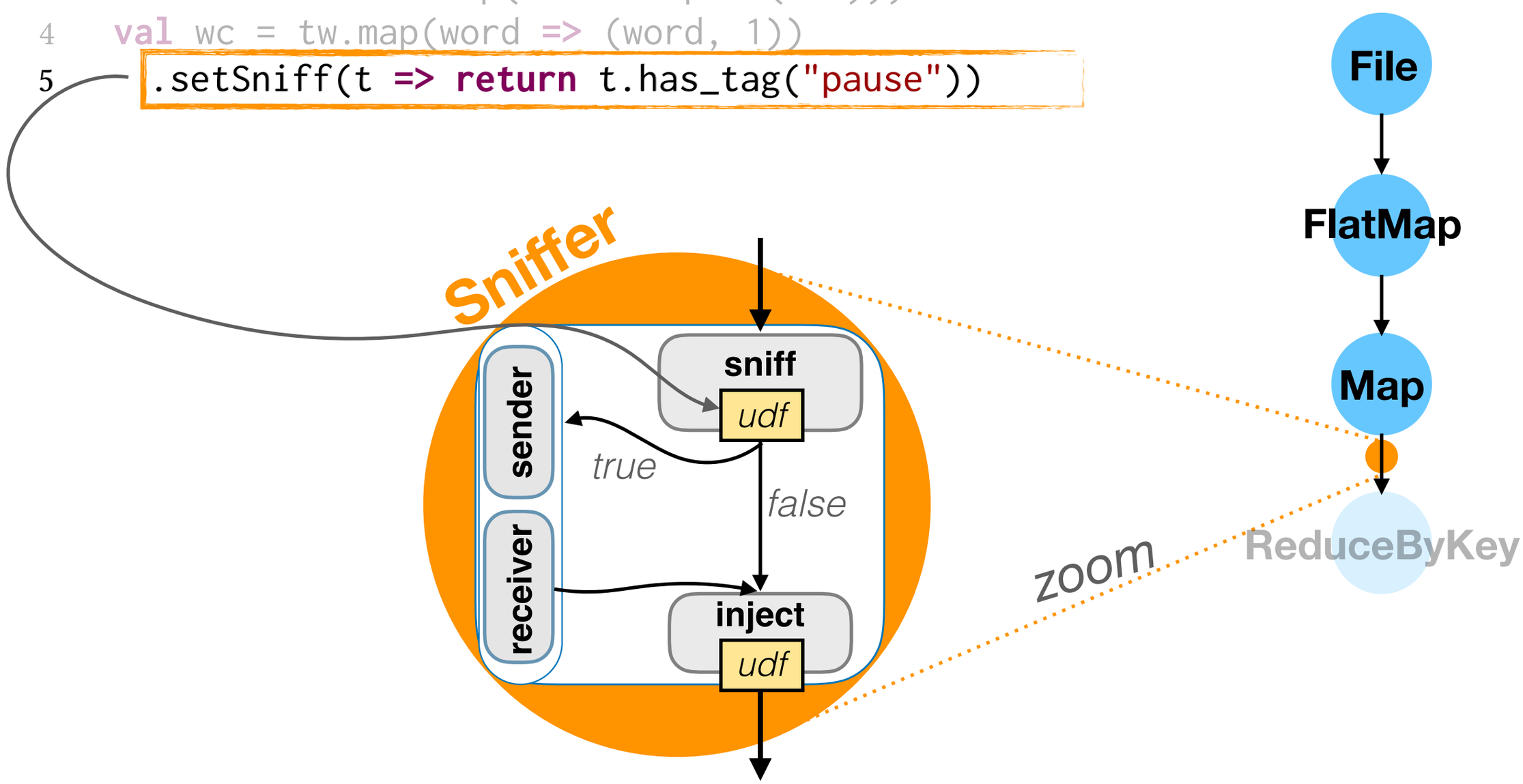
```
1   val ln = new RDDbug(spark.textFile(file))
2     .setTag(t => if (line_number(t) % 10000 == 0)
            t.add_tag("pause"))
3   val tw = ln.flatMap(l => l.split(" "))
4   val wc = tw.map(word => (word, 1))
5     .setSniff(t => return t.has_tag("pause"))
```



Sniffing Tuples

Code snippet: pausing every 10k tuples

```
1   val ln = new RDDbug(spark.textFile(file))
2     .setTag(t => if (line_number(t) % 10000 == 0)
            t.add_tag("pause"))
3   val tw = ln.flatMap(l => l.split(" "))
4   val wc = tw.map(word => (word, 1))
5     .setSniff(t => return t.has_tag("pause"))
```

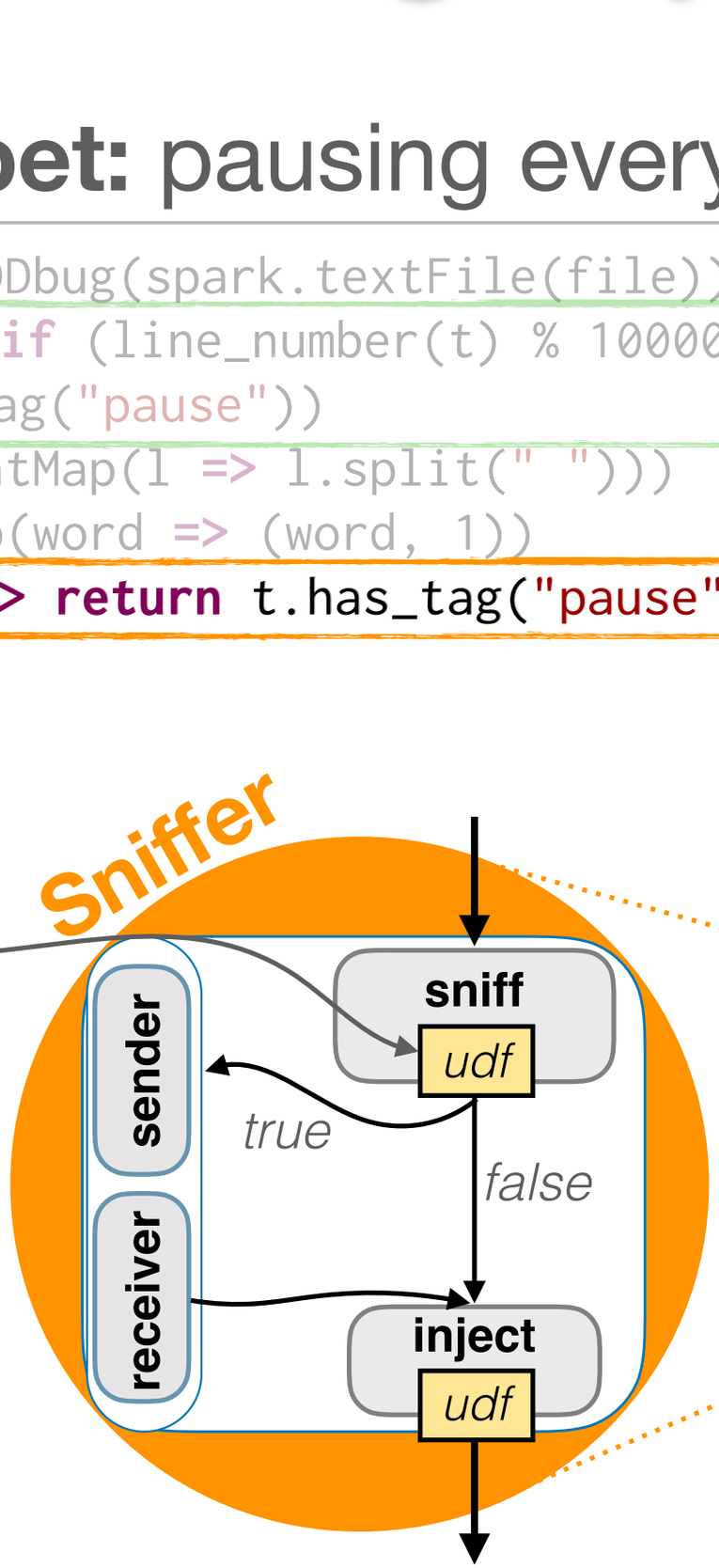


Sniffing Tuples

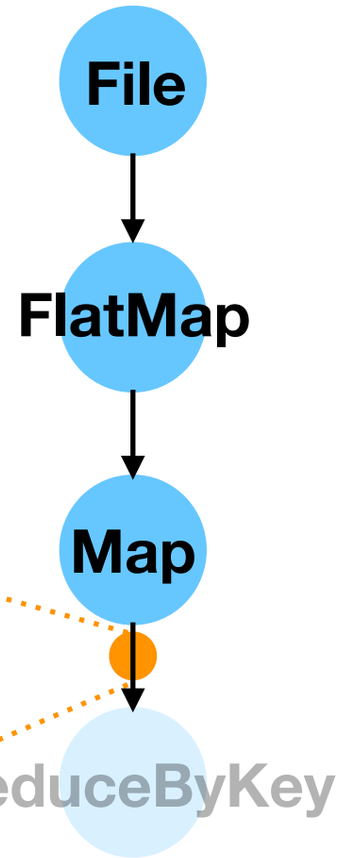
Code snippet: pausing every 10k tuples

```
1   val ln = new RDDbug(spark.textFile(file))
2     .setTag(t => if (line_number(t) % 10000 == 0)
            t.add_tag("pause"))
3   val tw = ln.flatMap(l => l.split(" "))
4   val wc = tw.map(word => (word, 1))
5     .setSniff(t => return t.has_tag("pause"))
```

Action
send-out
skip-tuple
job-halt

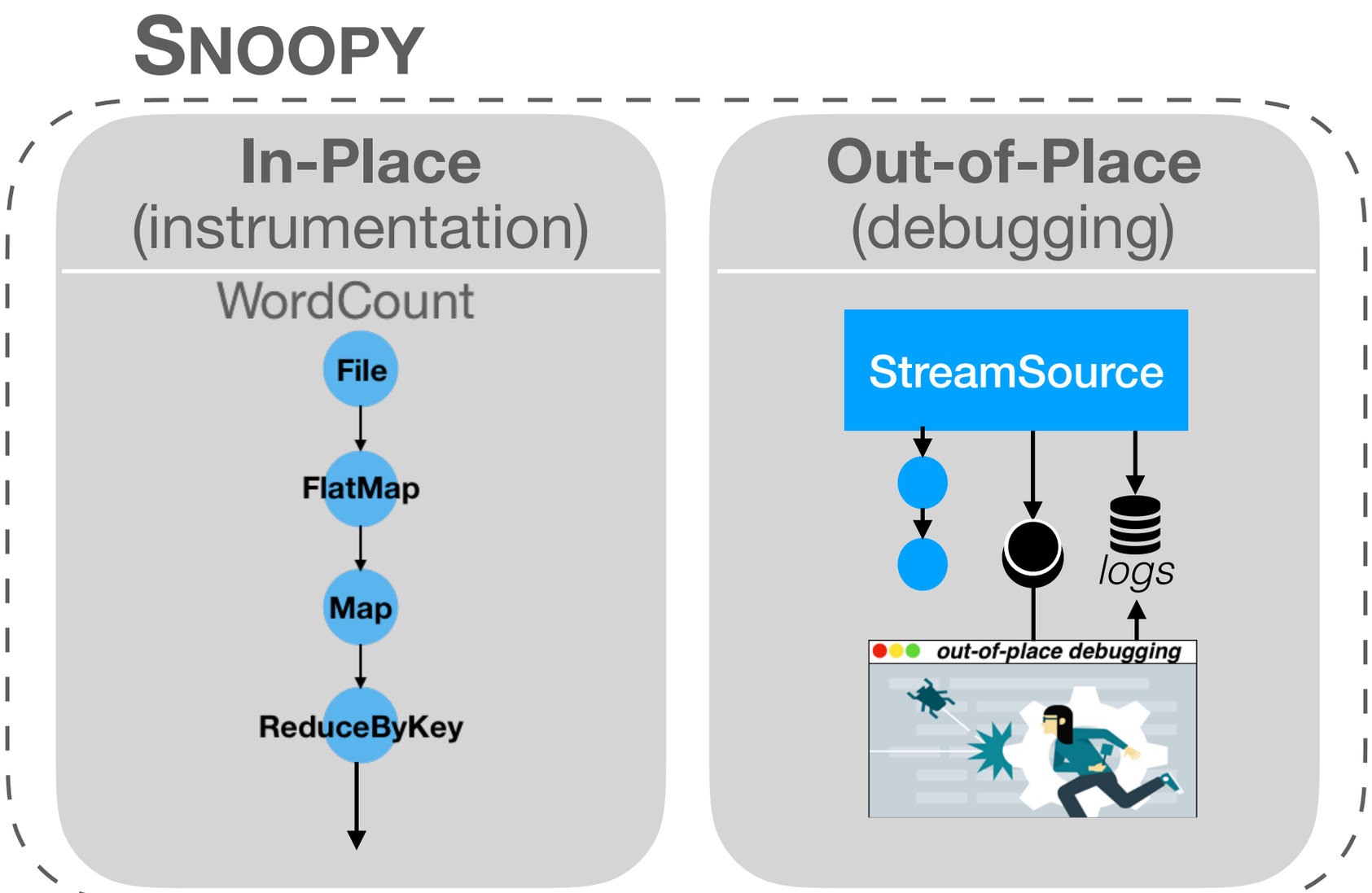


WordCount



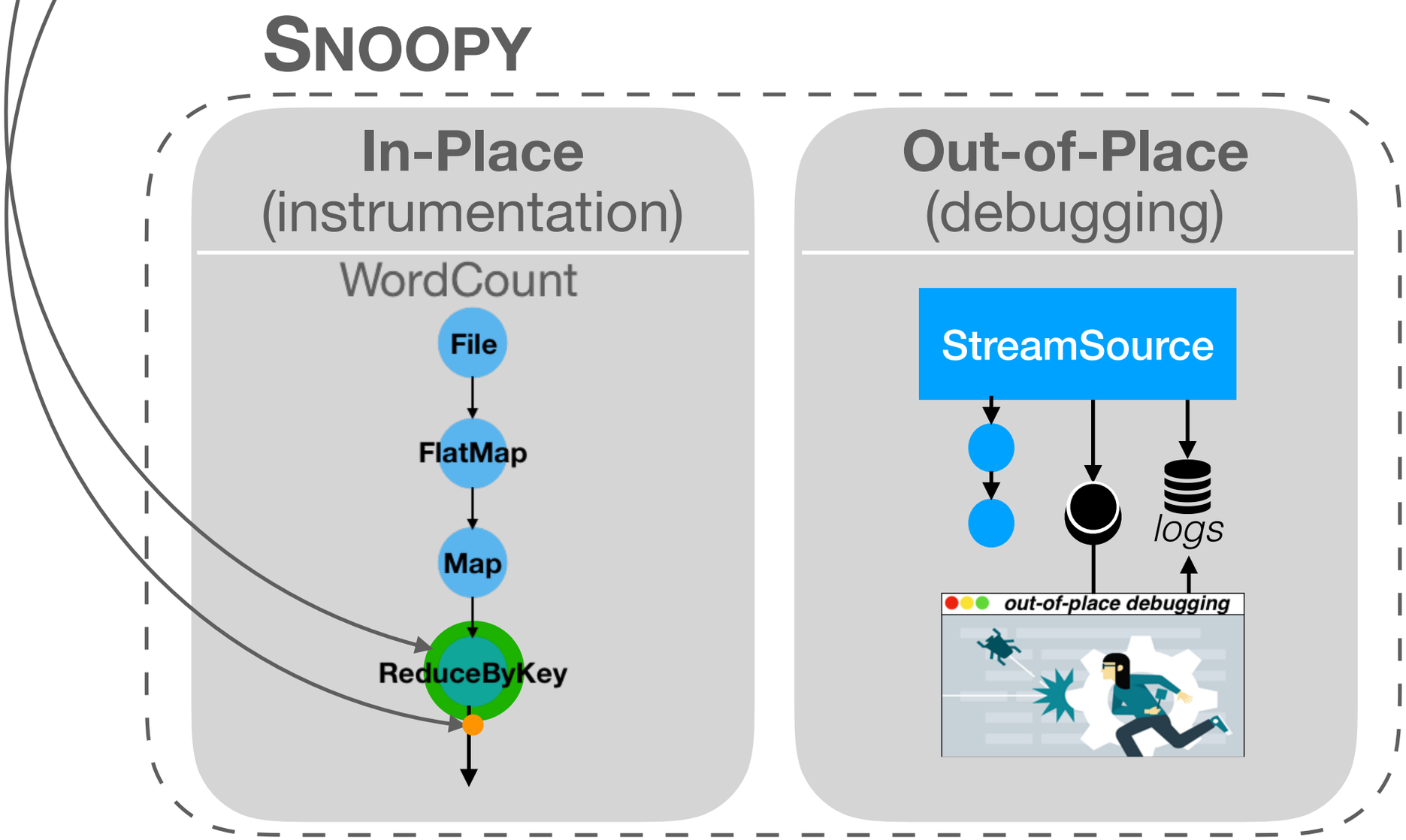
Debugging Tuples
— Perf. Bottleneck Example —

```
7   .setTag(t => t.add_tag("now-"+ System.currentTimeMillis()), PRE)
8   .setTag(t => {t_start = put_in_array(t.get_tag("now"))
9      if (System.currentTimeMillis() - t_start > 60000)
10       t.add_tag("alert")}, POST)
11  .setSniff(t => return t.has_tag("alert"))
```



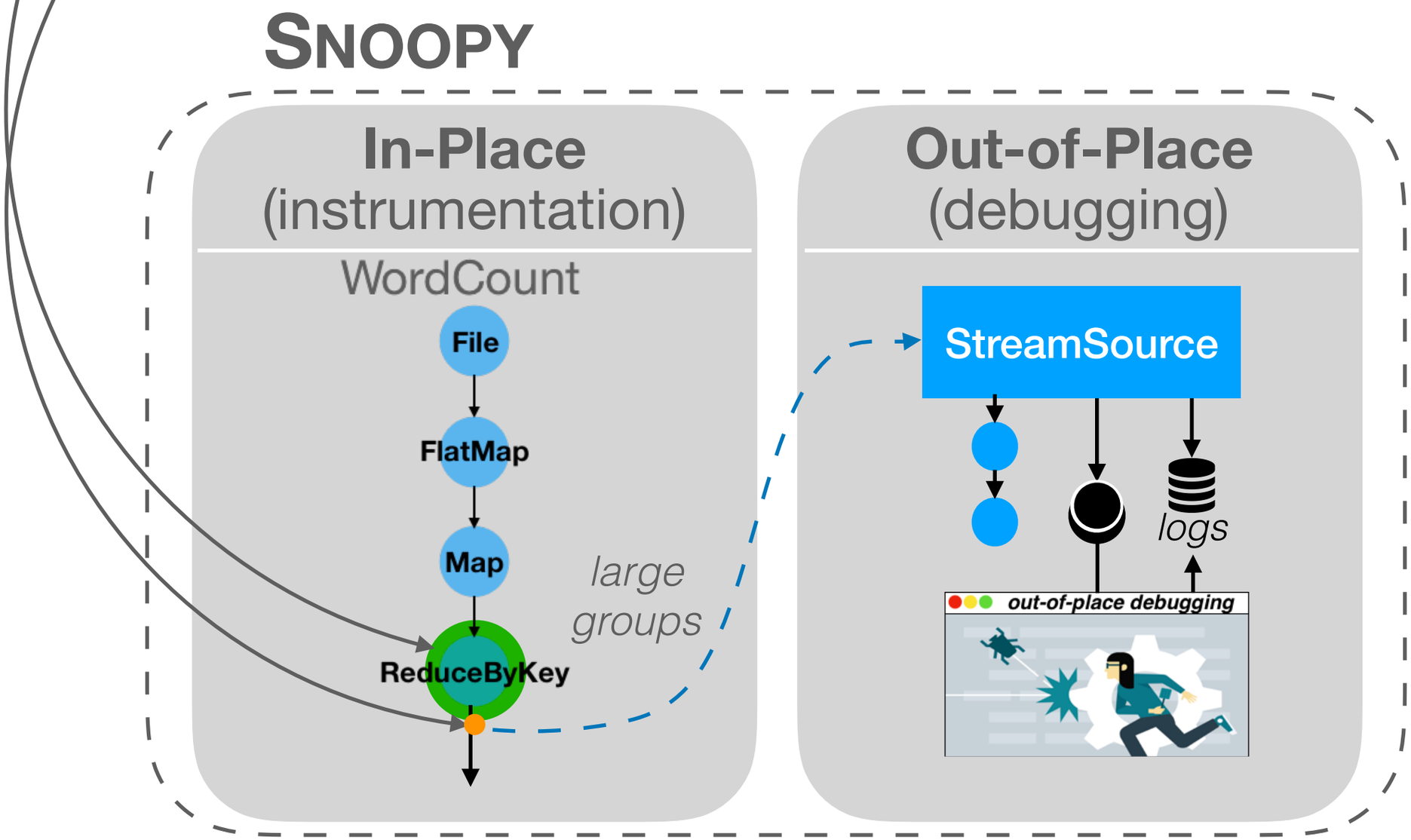
Debugging Tuples
— Perf. Bottleneck Example —

```
7   .setTag(t => t.add_tag("now-"+ System.currentTimeMillis()), PRE)
8   .setTag(t => {t_start = put_in_array(t.get_tag("now"))
9      if (System.currentTimeMillis() - t_start > 60000)
10       t.add_tag("alert")}, POST)
11  .setSniff(t => return t.has_tag("alert"))
```

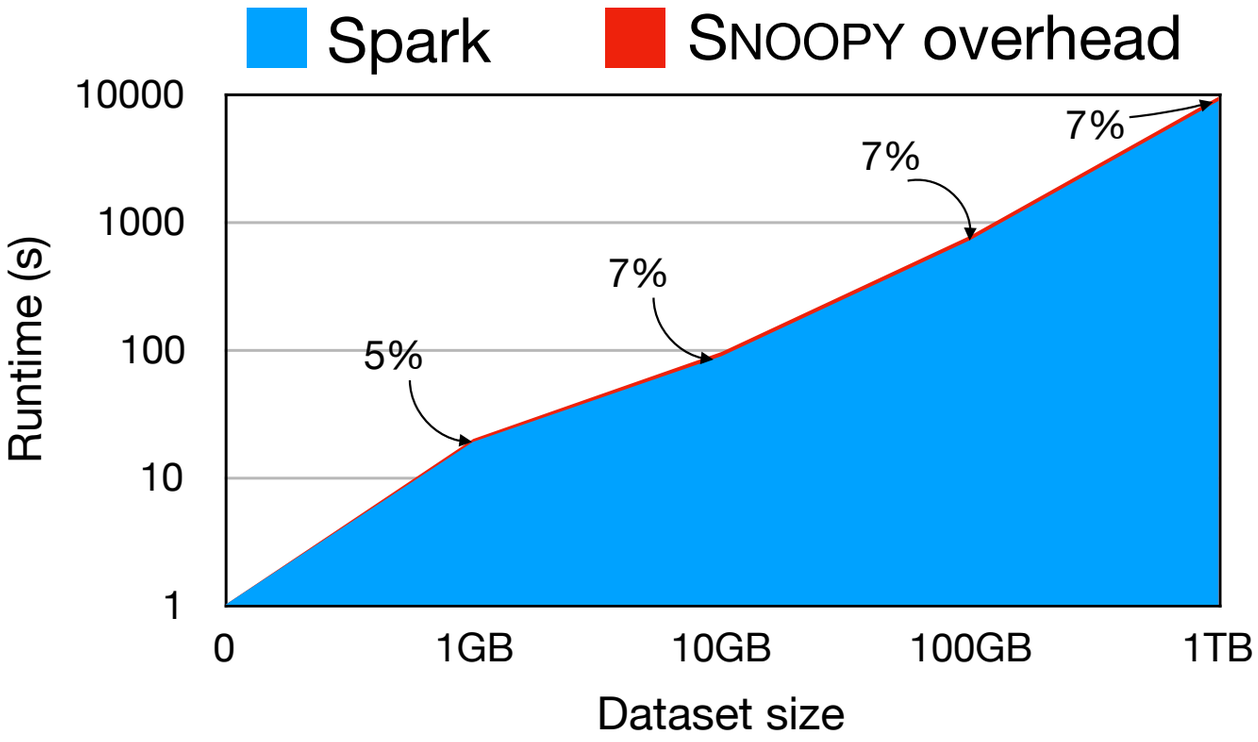


Debugging Tuples
— Perf. Bottleneck Example —

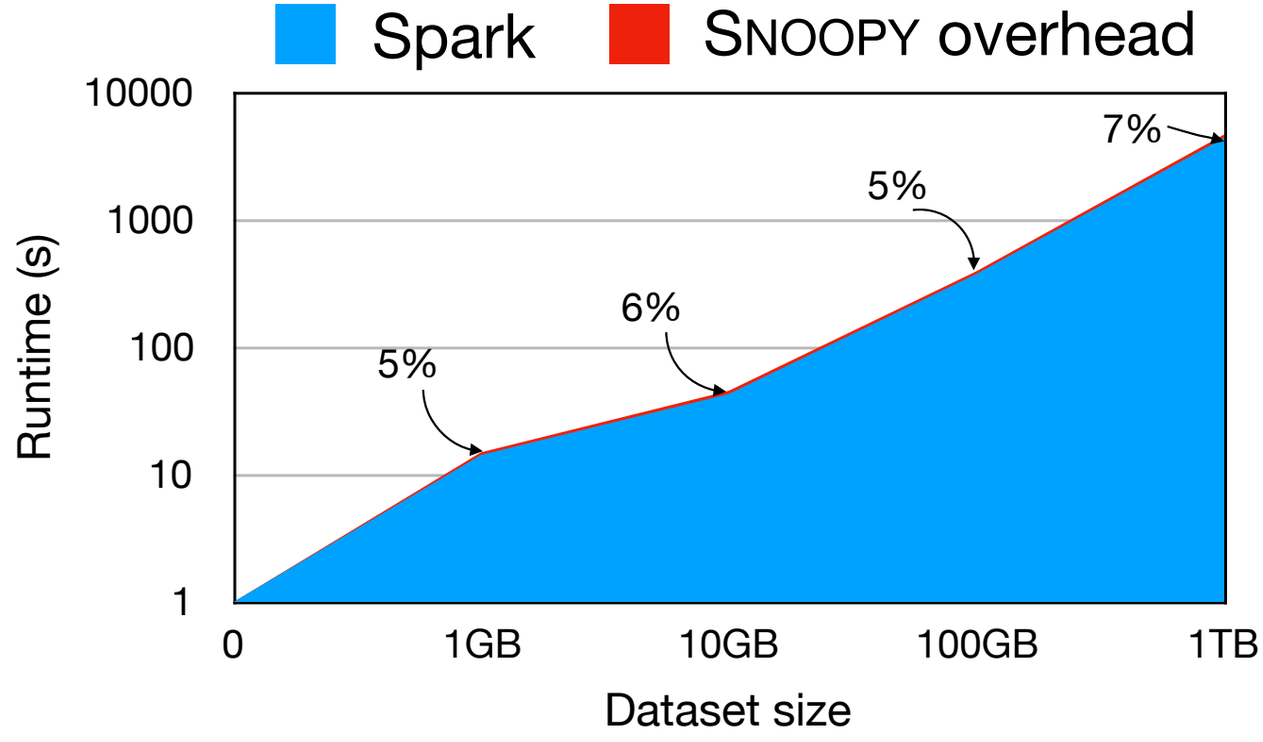
```
7   .setTag(t => t.add_tag("now-"+ System.currentTimeMillis()), PRE)
8   .setTag(t => {t_start = put_in_array(t.get_tag("now"))
9      if (System.currentTimeMillis() - t_start > 60000)
10       t.add_tag("alert")}, POST)
11  .setSniff(t => return t.has_tag("alert"))
```



SNOOPY Performance

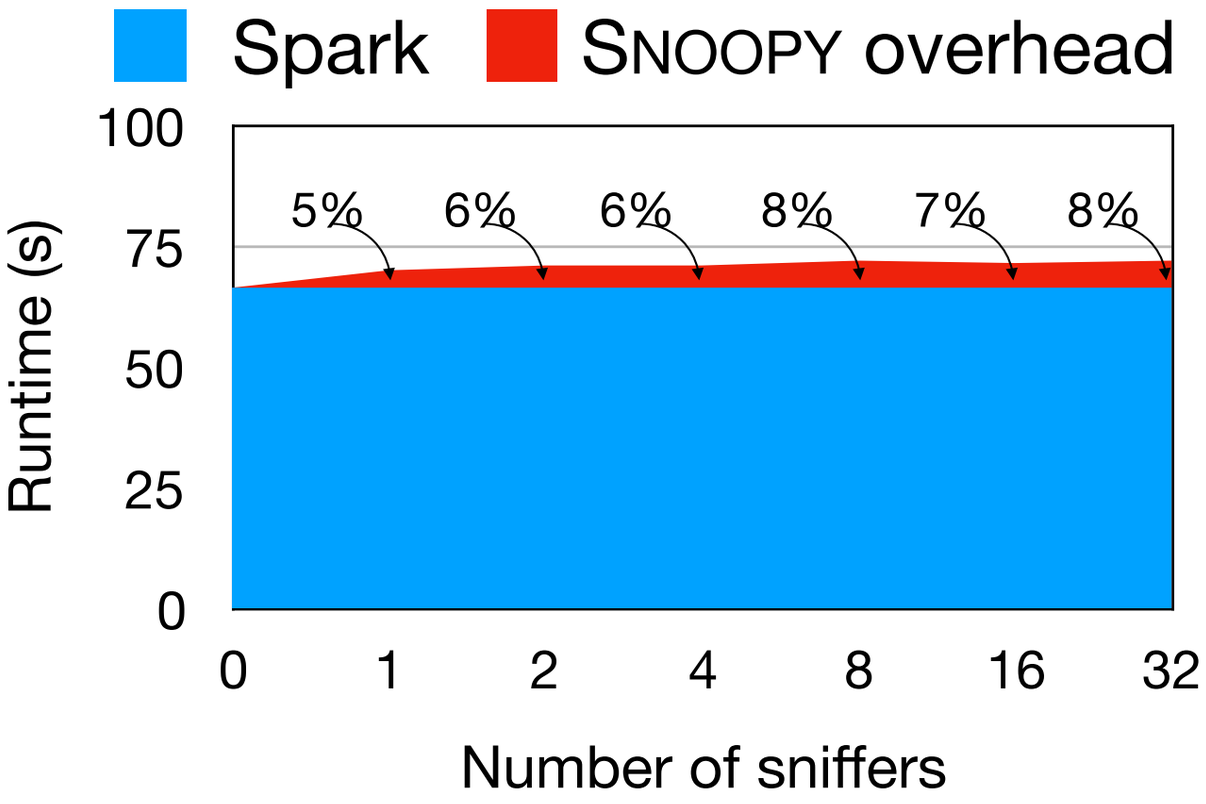


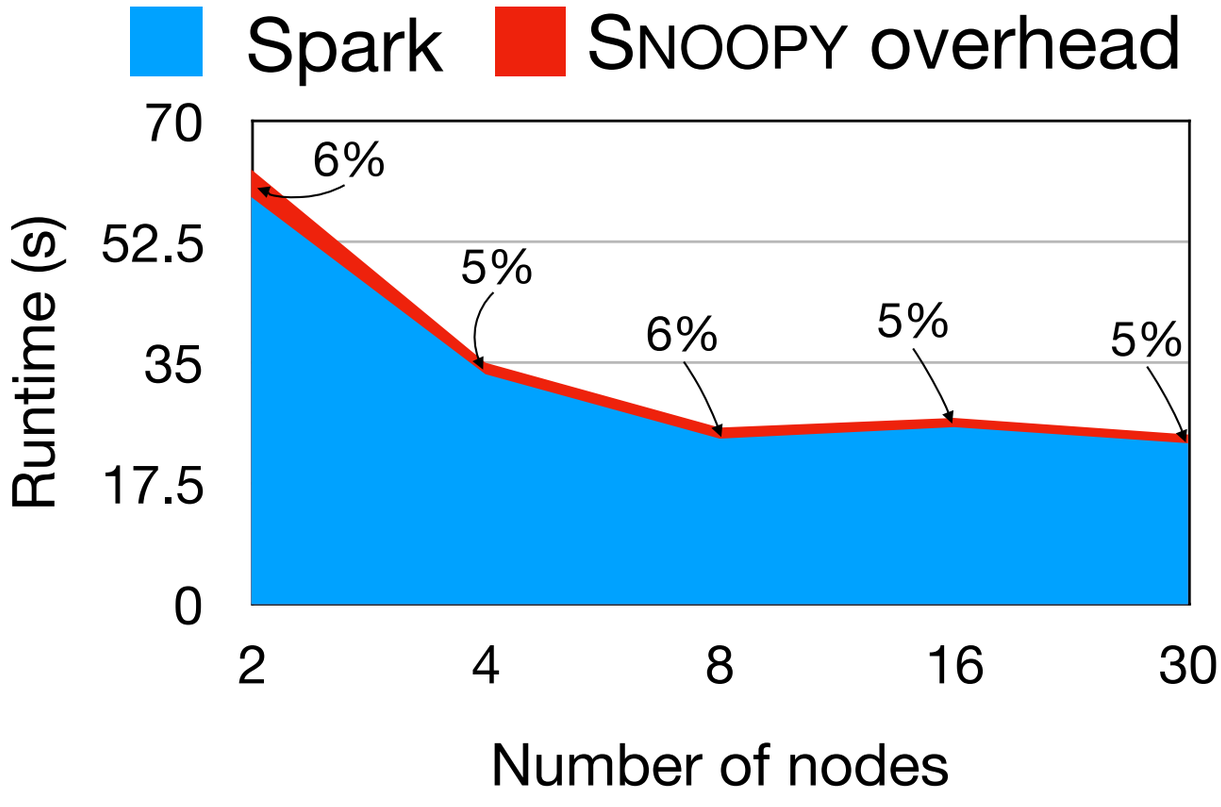
WordCount



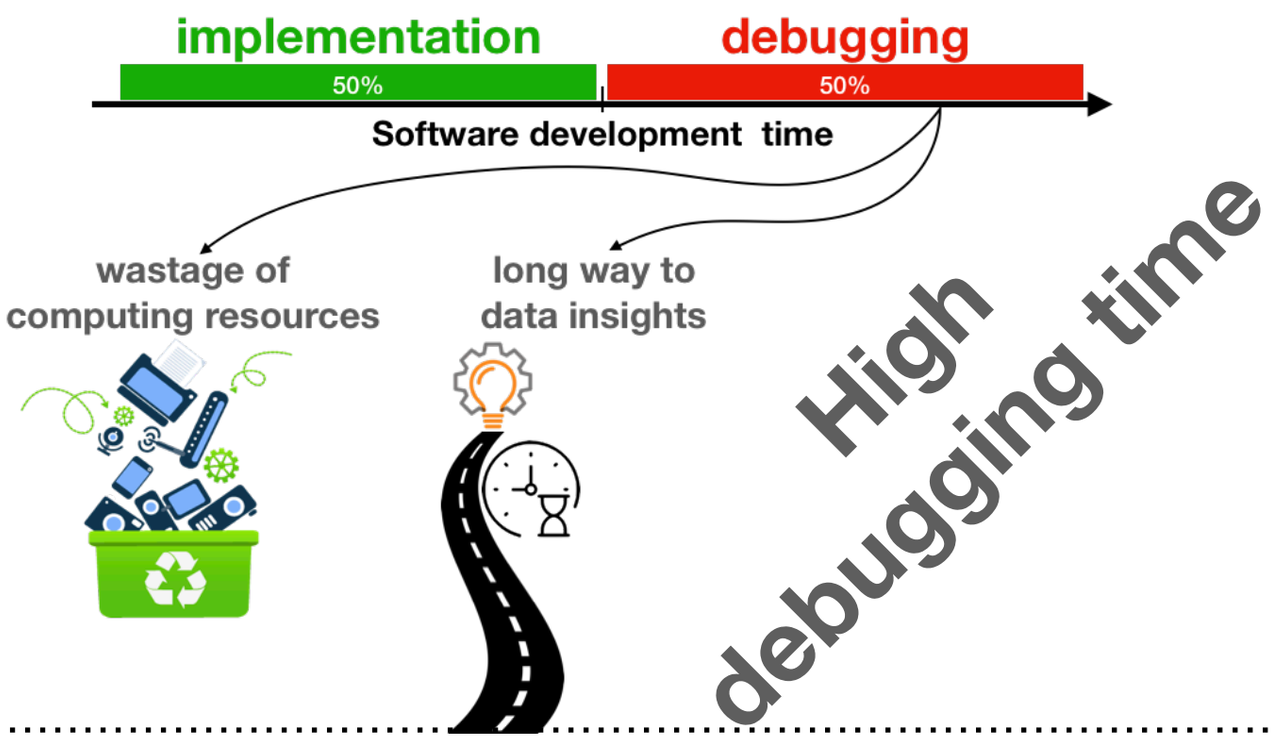
Join

Snoopy Scalability





TagSniff: Data Debugging for All

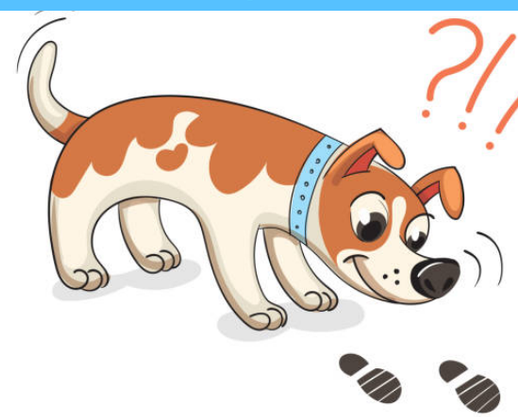


DebugTuple <tags, <tuple>>

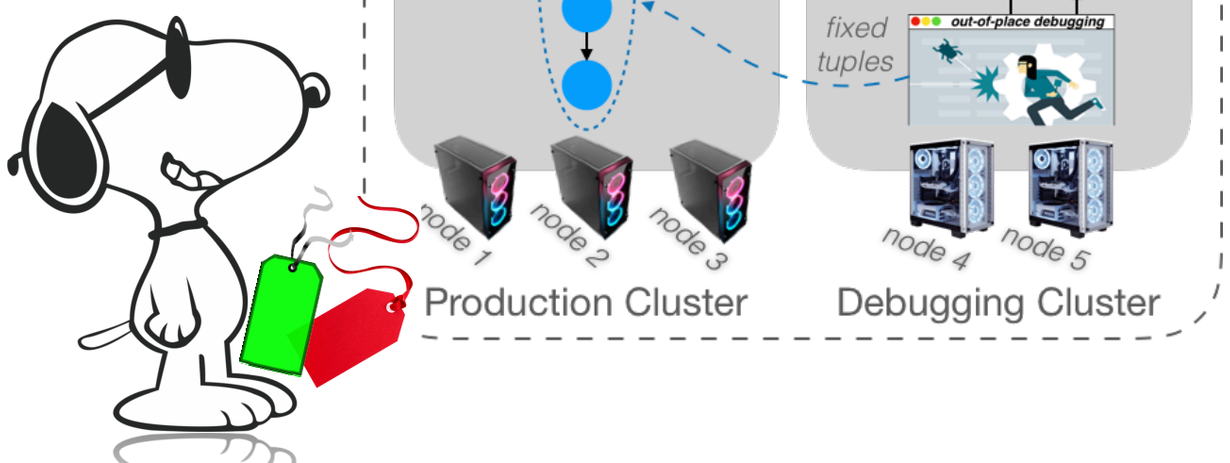
tag(f:tuple => tuple)

sniff(f:tuple => boolean)

TagSniff Model



SNOOPY



Very Low Overhead