EXTENDING SPARK WITH JAVA AGENTS

Jaroslav Bachorik
Adrian Popescu

Meet The Speakers

Jaroslav

- Backend engineer
- Hands-on experience with JVM
  - OpenJDK Reviewer
- Strong focus on performance
  - VisualVM Contributor
  - BTrace Maintainer and Lead Developer

Adrian

- Backend engineer
- 5+ years of experience in performance monitoring & modeling
- Focusing on tuning and optimization of Big Data apps

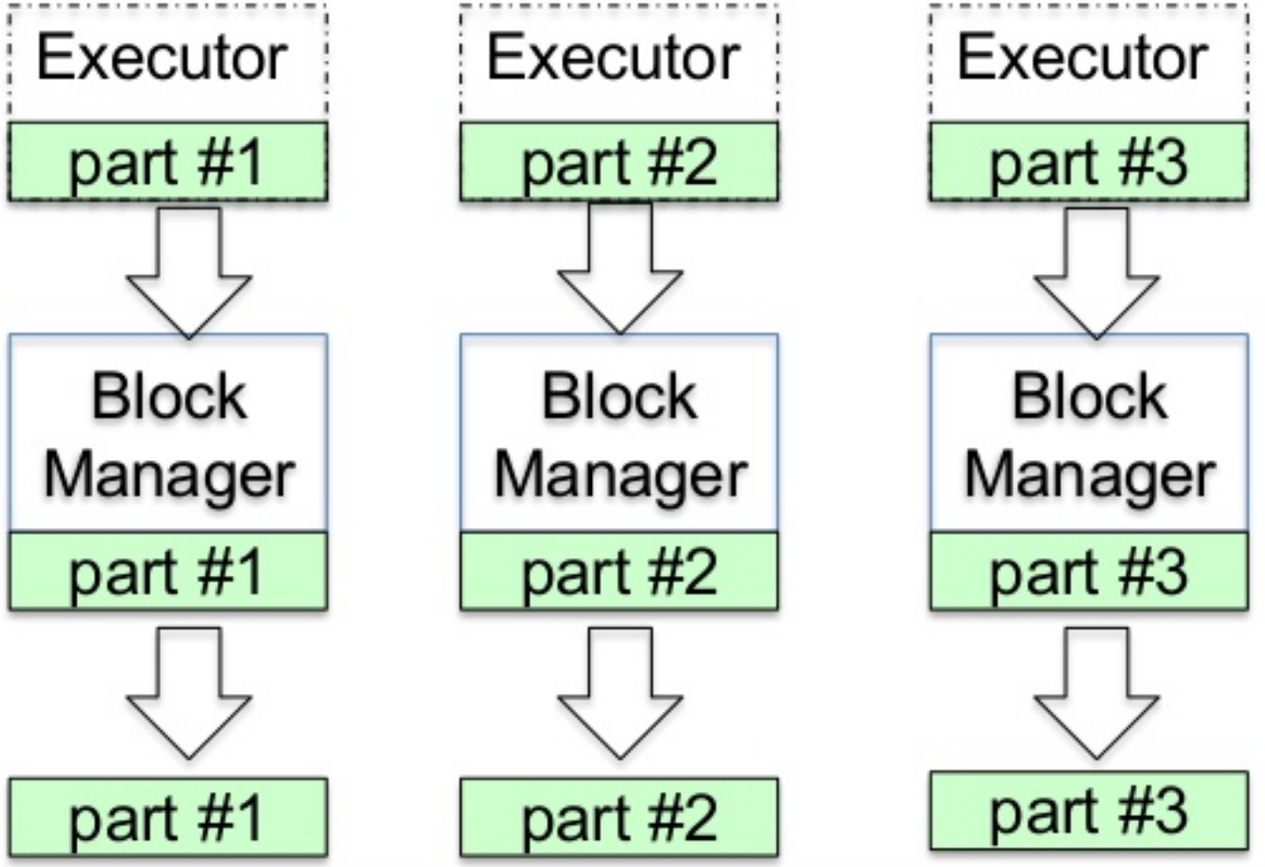
Unravel

Performance Intelligence for Big Data Stack

Caching in Spark

```
...
rdd.cache()
rdd.count()
...

rdd.reduceBy
Key((x,y) =>
x+y).count()
```



Avoiding re-evaluation for cached RDD blocks

Spark Caching Problem

- **RDD Caching**
  - Benefits: speedup, saved resources
  - Too much caching may waste resources

- **Spot the opportunity**
  - Which RDDs to cache (for most benefit)?
  - How to prioritize when memory is scarce?
  - Where to cache?
  - Benefit depends on memory state

- http://unraveldata.com/to-cache-or-not-to-cache/

Challenging to decide **when, what,** and **where** to cache

Algorithm

1. **Find** stages that share RDDs

2. **Cached RDDs**: Measure benefit
   Block *hitRate & time saved* for every Stage

3. **RDDs not cached**: Potential benefit
   Approximate time saved & RDD storage size

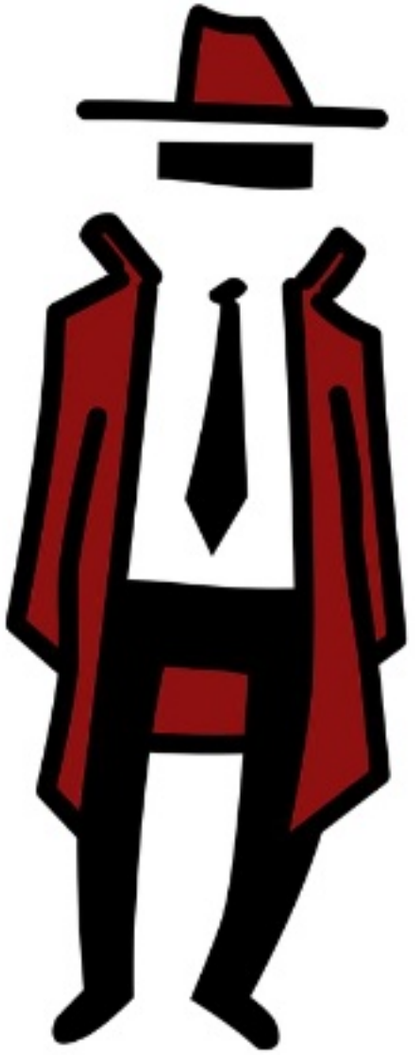
Insufficient visibility into Spark structures

e.g., only block info in **BlockStatusListener**

4. **Suggest** which RDDs to cache

Need to collect additional metrics

Meet Java Agent



Meet Java Agent

Launcher

```
java -javaagent:myfancyagent.jar=<args>
```

- Runs JVM with an agent
- Agent jar
  - agent class
  - required manifest entries
- Arguments
  - any string understood by agent class
- Multiple agents can be specified

```
Premain-Class: com.myorg.agents.FancyAgent

Can-Redefine-Classes: true

Can-Retransform-Classes: true
```

Meet Java Agent

Launcher

```
java -javaagent:myfancyagent.jar=<args>
```

FancyAgent.java

```
public static void premain(String args, Instrumentation inst) {
```

- Executed at JVM startup
- **Instrumentation** provides access to JVM internals

META-INF/manifest.mf

```
Premain-Class: com.myorg.agents.FancyAgent
Can-Redefine-Classes: true
Can-Retransform-Classes: true
```

Meet Java Agent

Launcher

```
java -javaagent:myfancyagent.jar=<args>
```

FancyAgent.java

```
public static void premain(String args, Instrumentation inst) {
    // do the magic stuff here
}
```

META-INF/manifest.mf

```
Premain-Class: com.myorg.agents.FancyAgent
```

- Agent class
- Must be present in the agent jar

Meet Java Agent

Launcher

```
java -javaagent:myfancyagent.jar=<args>
```

FancyAgent.java

```
public static void premain(String args, Instrumentation inst) {
    // do the magic stuff here
}
```

META-INF/manifest.mf

Transform classes upon load

```
Can-Redefine-Classes: true
Can-Retransform-Classes: true
```

Meet Java Agent

Launcher

```
java -javaagent:myfancyagent.jar=<args>
```

FancyAgent.java

```
public static void premain(String args, Instrumentation inst) {
    // do the magic stuff here
```

- Agent capabilities and settings
  - retransform classes
  - redefine classes
  - native method prefix

https://docs.oracle.com/javase/7/docs/api/java/lang/instrument/package-summary.html

```
Can-Retransform-Classes: true
```

Class Transformers

```
class SpecialLibraryClass {
  public void doStuff() {
    System.out.println(
        "Doing stuff #" + id(System.currentTimeMillis())
    );
  }
  private int id(int input) {
    return input % 42;
  }
}
```

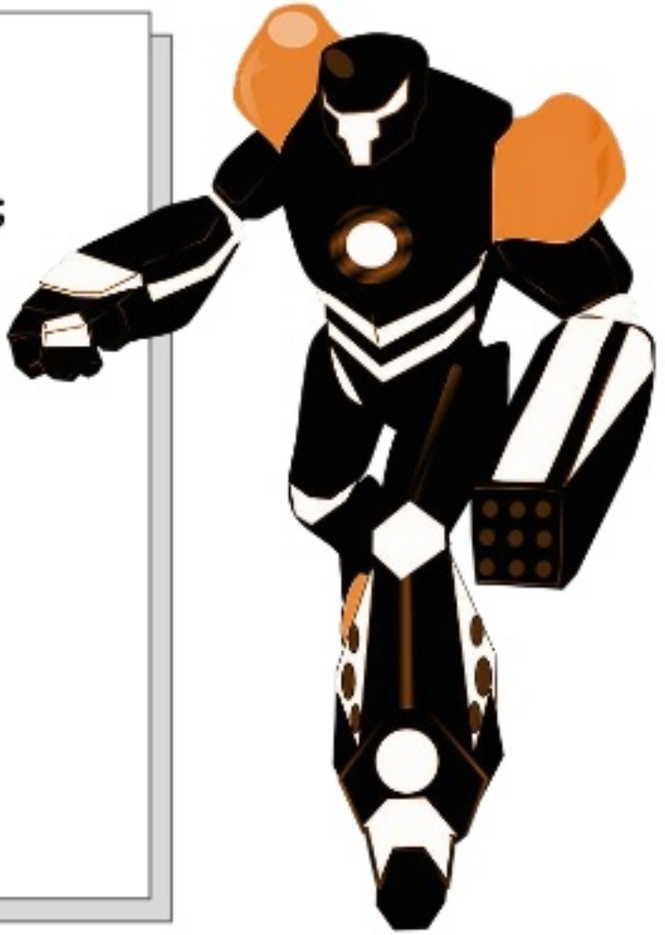
Private method in 3rd party package private method.

Patch and recompile? Reflection?

http://pixabay.com

Class Transformers

```
class SpecialLibraryClass {
  public void doStuff() {
    System.out.println("Doing stuff #" + id(System.currentTimeMillis()));
  }
  private int id(int input) {
    AgentCallbacks.onEnter("SLC", "id", "input = " + input);
    int tmp = input % 42;
    AgentCallbacks.onReturn("SLC", "id", tmp);
    return tmp;
  }
}
```



http://pixabay.com

Class Transformers

```
java.lang.instrument.ClassTransformer

    byte[] transform(ClassLoader l, String name, Class<?> cls,
                     ProtectionDomain pd, byte[] classfileBuffer)
```

- Inspect and modify class data
  - without recompilation
  - even on-the-fly
- Powerful
  - introduce callbacks
  - patch problematic code
- Not a simple task
  - class data structure
  - constant pool
  - stack frame maps

http://wikipedia.org

BTrace

https://github.com/btraceio/btrace

btrace.io

- Open source dynamic instrumentation framework for JVM
- Java annotations based DSL for class transformations
- Optimized for high performance
- Used eg. in VisualVM (http://visualvm.github.io)

Spark Config Tuning

- Intercept **`SparkContext`** instantiation at Spark Driver
- Modify the associated **`SparkConf`** object
  - before **`SparkContext`** is fully instantiated
- Any Spark property can be changed
  - eg. "`spark.executor.cores`" based on system load
- Modified **`SparkConf`** will be applied to the current job
  - and distributed to to all executors

Config Tuning Transformation

```
@OnMethod(clazz = "org.apache.spark.SparkContext",
          method = "<init>",
          location = @Location(Kind.RETURN)
)
public static void sparkContextConstructor(@Self SparkContext scObj) {
    SparkConf conf = scObj.conf();
    if (conf == null)
        return;
    int recommendedCores = AdviceSystem.getRecommendedCores(scObj);
    conf.set("spark.executor.cores", recommendedCores);
}
```

Agents on Spark?

http://apache.org
http://pixabay.com

Agents on Spark!

- Spark is a JVM application
- Any Java agent can be used with Spark

http://apache.org
http://pixabay.com

Register JVM Agent for Spark

```
java -javaagent:myfancyagent.jar=<args>
```

Spark configuration parameters

```
spark.executor.extraJavaOptions=
    "-javaagent:/opt/myagent/myagent.jar=opt.load:true"
spark.driver.extraJavaOptions=
    "-javaagent:/opt/myagent/myagent.jar=track.conf:true"
```

Store the options in **spark-defaults.conf** to apply system-wide.

http://arthurandsage.blogspot.cz/

Let's Put It All Together!

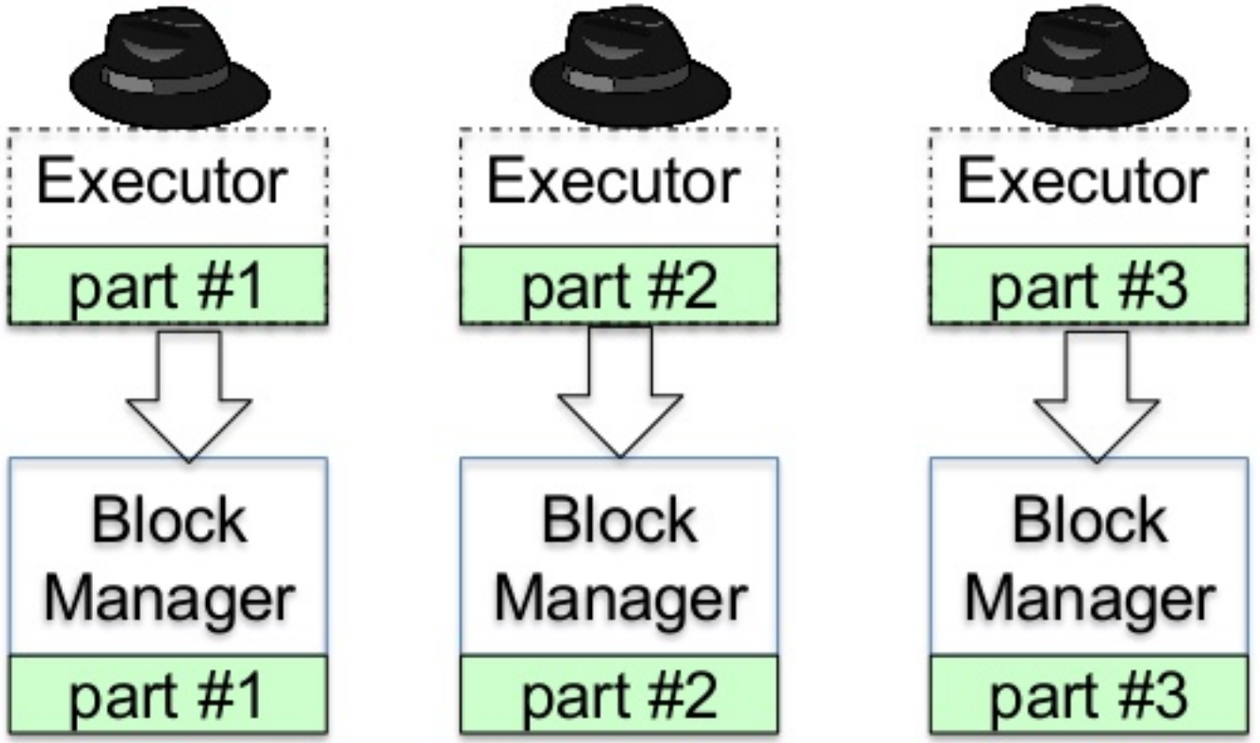
Identify Required Data

```
...
rdd.cache()
rdd.count()
...

rdd.reduceBy
Key((x,y) =>
x+y).count()
...
```

1. Evaluation

2. Caching



Agents allow:

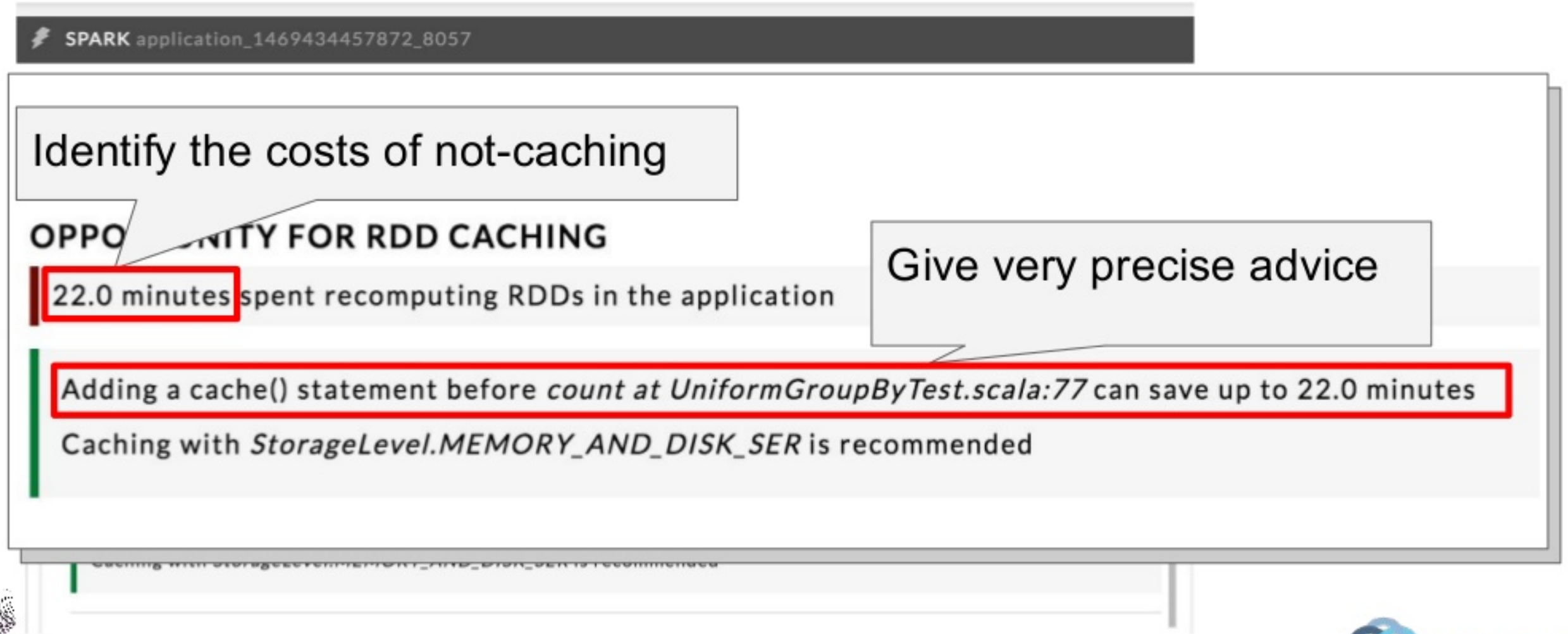
- **Access to Spark internal structures**
- **Collecting**: RDD block usage, cache hit, potential benefit
- **Run algorithm and suggest** which RDDs to cache

Transform Classes

```
def get(blockId: BlockId): Option[BlockResult] = {
  val local = getLocal(blockId)
  if (local.isDefined) {
    logInfo(s"Found block $blockId locally")
    return local
  }
  val remote = getRemote(blockId)
  if (remote.isDefined) {
    logInfo(s"Found block $blockId remotely")
    return remote
  }
  None
}
```

```
def get(blockId: BlockId): Option[BlockResult] = {
  AgentRuntime.useBlock(blockId, currentStage)
  val local = getLocal(blockId)
  if (local.isDefined) {
    logInfo(s"Found block $blockId locally")
    return local
  }
  val remote = getRemote(blockId)
  ...
  ...
}
```

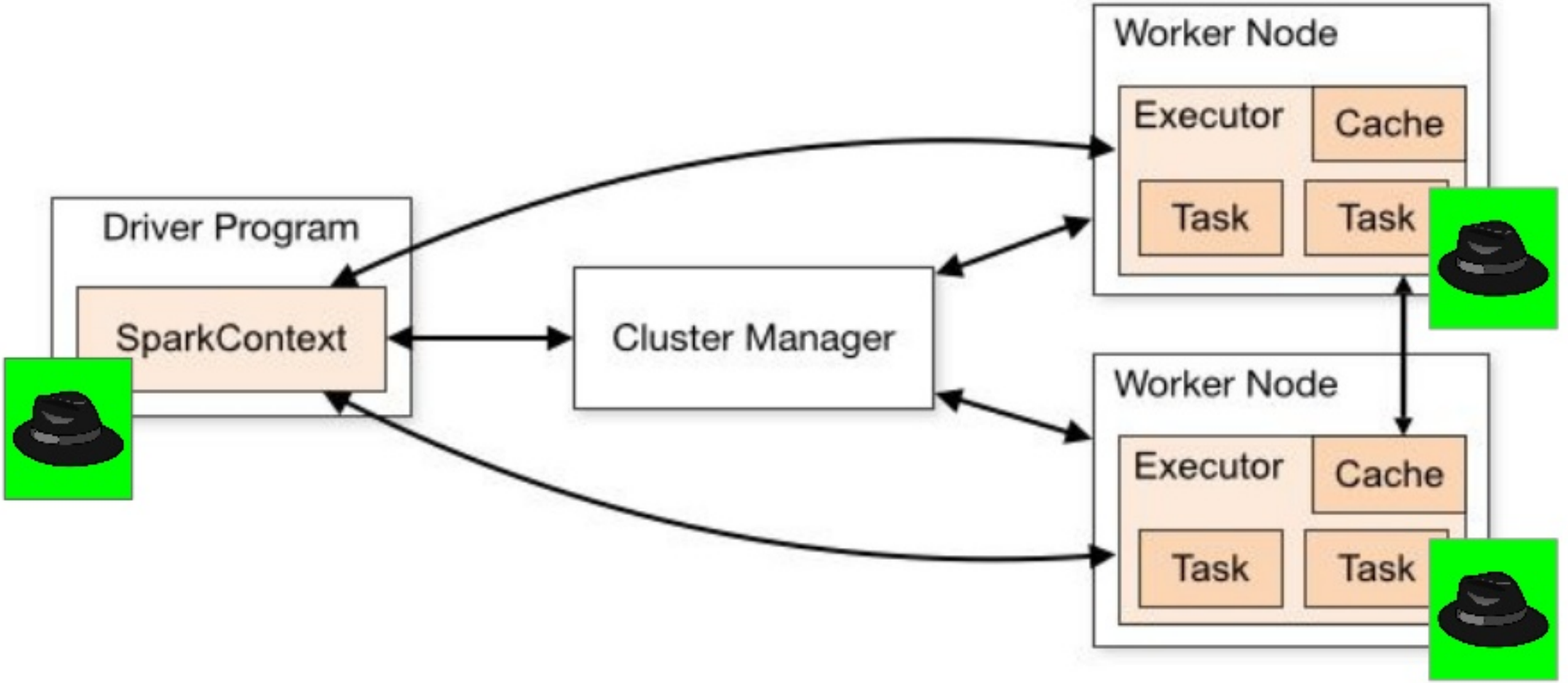
Unravel Spark Cache Events



unravel

Put It On Cluster

Agents In Cluster



http://http://spark.apache.org/

Distributing Agents for Spark

Make sure all nodes have the latest and same copy of agent binaries

- Manually copy agents to all nodes
- Script that copy process

- Put that script in CRON!

Yarn Localization Service

- Built-in resource distribution system
- Cluster resources
  - linked into executor working directories
  - kept up-to-date
- Supported resource types
  - files
  - archives
- https://spark.apache.org/docs/1.5.2/running-on-yarn.html

Takeaways

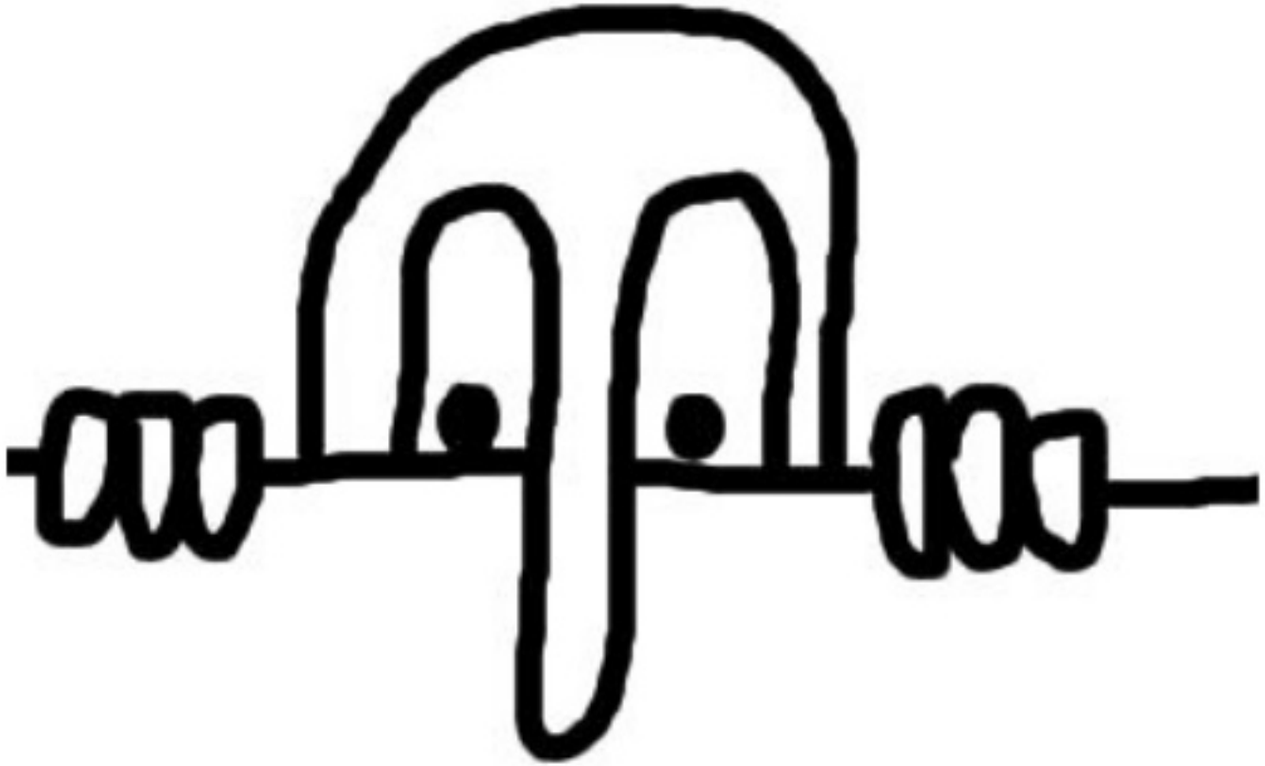
- Sometimes even rich set of Spark metrics, logs and events is not sufficient
- Java agents are powerful tool
  - extend application beyond the original intent
  - hot-patch problems or enhancements
  - may be removed when not needed
- Try Unravel to see the power of Java agents unleashed

Resources

- To simplify application and operations management with Spark, try out Unravel
  - http://unraveldata.com/free-trial/
- Start with Java agents and try out BTrace
  - https://github.com/btraceio/btrace
  - contributors most sincerely welcome!

- Spark Configuration
  - http://spark.apache.org/docs/latest/configuration.html
  - https://spark.apache.org/docs/1.5.2/running-on-yarn.html
- Java Agents
  - https://docs.oracle.com/javase/7/docs/api/java/lang/instrument/package-summary.html
  - http://zeroturnaround.com/rebellabs/how-to-inspect-classes-in-your-jvm/

Q&A



Unravel Free Trial: http://unraveldata.com/free-trial/

THANK YOU.

Jaroslav Bachorik, jaroslav@unraveldata.com
Adrian Popescu, adrian@unraveldata.com