

Variable Tracker

Preparation for Bachelor project
SCG seminar project presentation

Overview

- Bachelor/seminar project
- Implementation
 - Reflectivity
 - How is it done?
 - Caching + Optimization
 - Speed comparison
- Demo

Bachelor project

- Give a man a dynamically typed language, will he write dynamically typed code?

```
|a|  
a := 42.  
a := 'hello world'.  
a := Dictionary new.
```

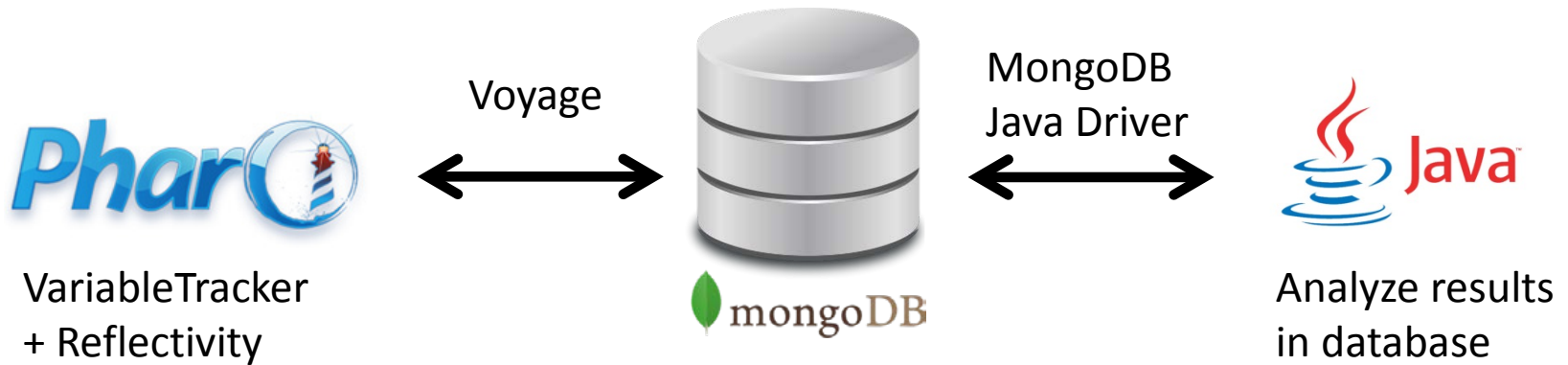
- Do developers need/use this behaviour?

```
|a b c|  
a := 42.  
b := 'hello world'.  
c := Dictionary new.
```

Seminar project

- Program to monitor how types change during execution
- Data is stored to MongoDB
- Program to analyze the results from the database

Interaction



Overview

- Bachelor/seminar project
- **Implementation**
 - Reflectivity
 - How is it done?
 - Caching + Optimization
 - Speed comparison
- Demo

Reflectivity

«Reflectivity is a tool to annotate AST nodes with metalinks. A metalink corresponds to a message sent to an arbitrary object.»

Example:

```
MyClass methods do: [ :method |  
    method ast  
        forAllNodes: [ :node | node isAssignment ]  
        putAfter: [ RFMetalink fromExpression:  
            'Transcript crShow: ``variable written``' ];  
        installWrapper ]
```

Variable written

Install wrapper

Add this code

My code

```
addWrapper: aMethod
```

Variable written

```
  aMethod ast
```

```
    forAllNodes: [ :node | node isAssignment ]
```

```
    putAfter: [ :node | RFMetalink fromExpression:
```

```
      '(VariableInformation new',
```

```
      ' variable: ', node variable name asString,
```

```
      ' name: ''', node variable name asString, ''',
```

```
      ' isInstanceVariable: ', node variable isInstance asString,
```

```
      ' class: self class name asString',
```

```
      ' method: thisContext method selector asString',
```

```
      ' isClassSide: self class isClassSide asString',
```

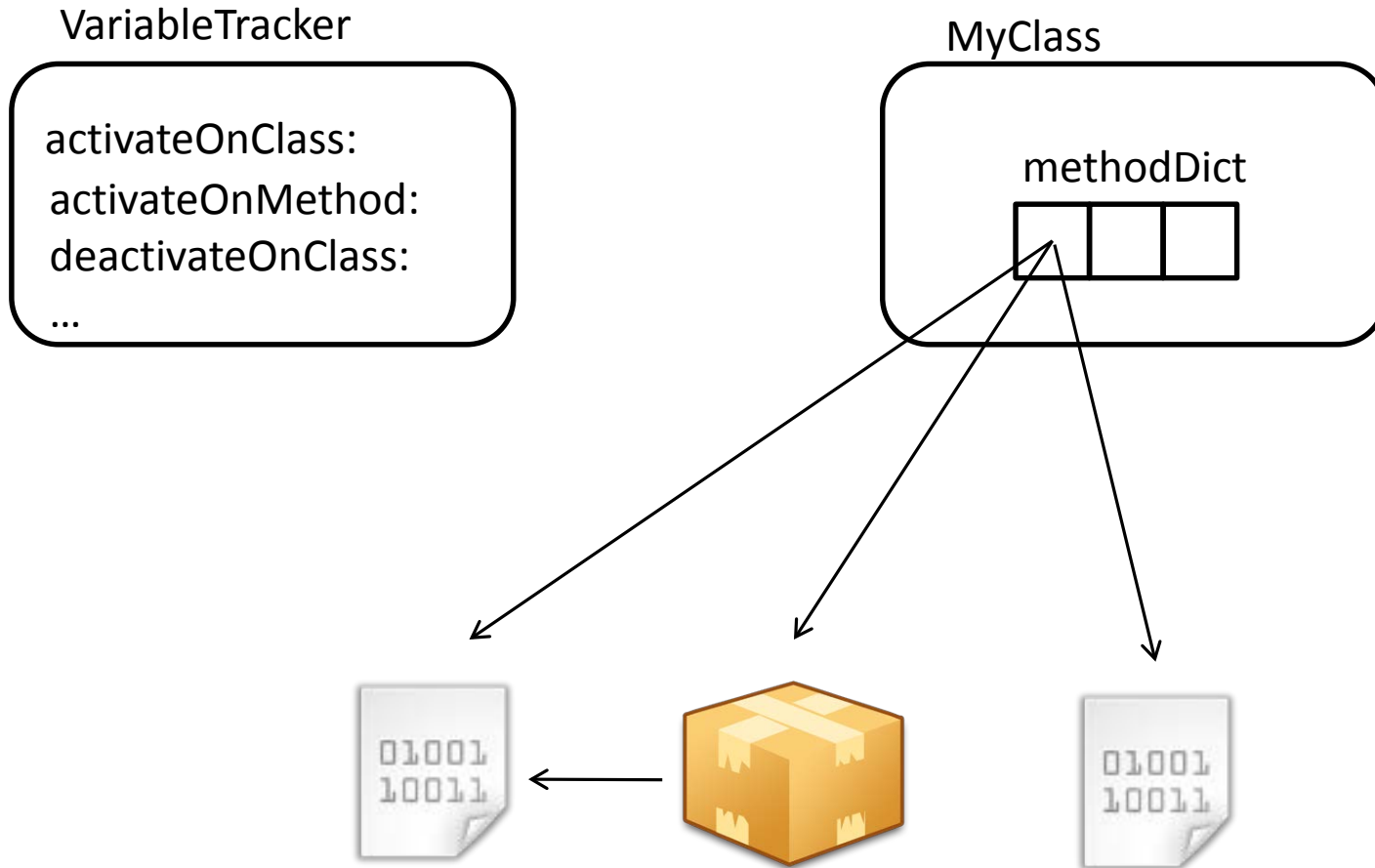
```
    ') save' ];
```

```
installWrapper.
```

Install wrapper

Save VariableInformation object
to DB

How is it done?



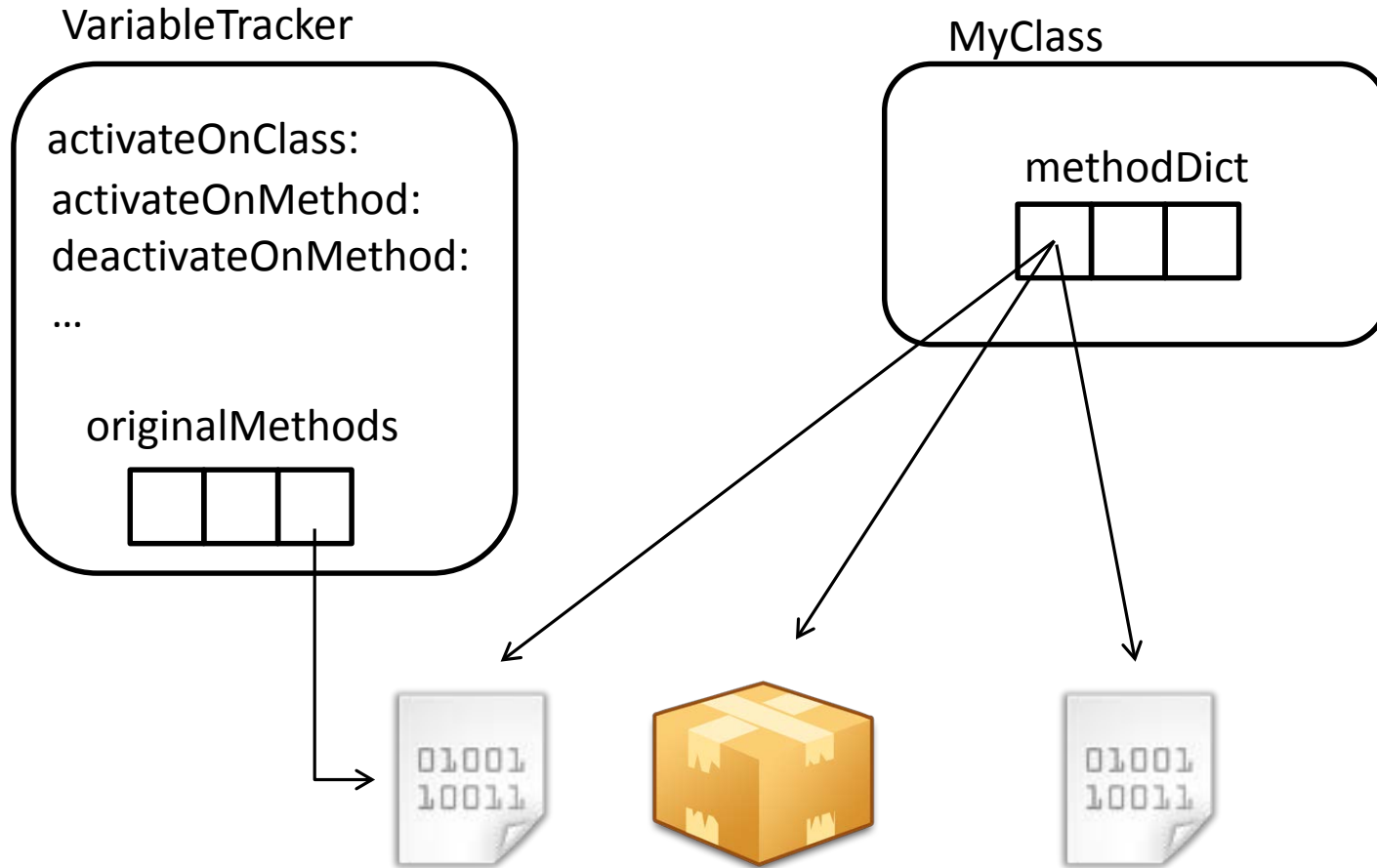
Speed comparison

	Duration
Without tracker	
First start	311ms
Next starts	~264ms
VariableTracker	
First start	16.5s
Next starts	~11s

Action:
Starting Nautilus browser

	Variables written
First start	~800
Next starts	~160

Caching + Optimization



Speed comparison

	Duration
Without tracker	
First start	311ms
Next starts	~264ms
VariableTracker without caching	
First start	16.5s
Next starts	~11s
VariableTracker with caching	
First start	1.496s
Next starts	~413ms

Action:
Starting Nautilus browser

	Variables written
First start	~800
Next starts	~160

First call: **11x** faster
Next calls: **26x** faster

Overview

- Bachelor/seminar project
- Implementation
 - Reflectivity
 - How is it done?
 - Caching + Optimization
 - Speed comparison
- **Demo**