

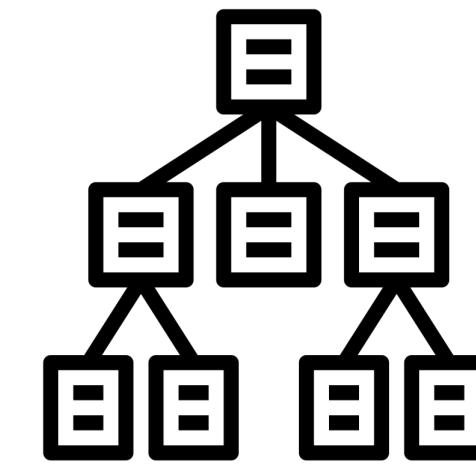
White-box Analysis for Modeling and Debugging the Performance of Configurable Systems

Thesis Proposal

Miguel Velez

**Most software is
configurable**

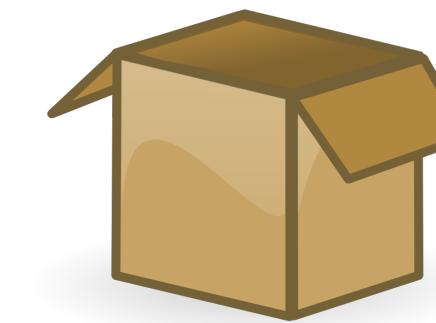
Configurable System



Indexing



Encryption



Compression

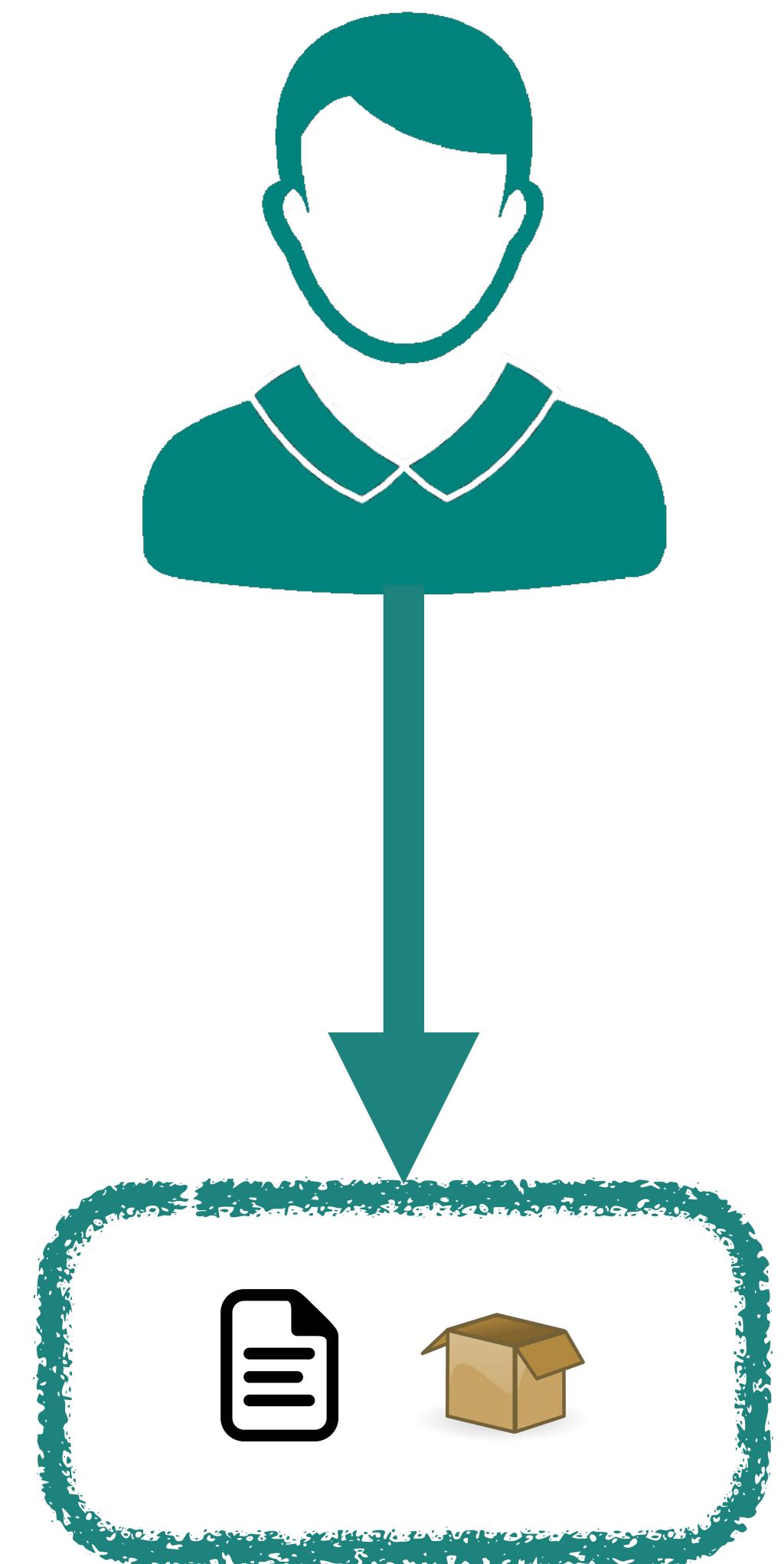
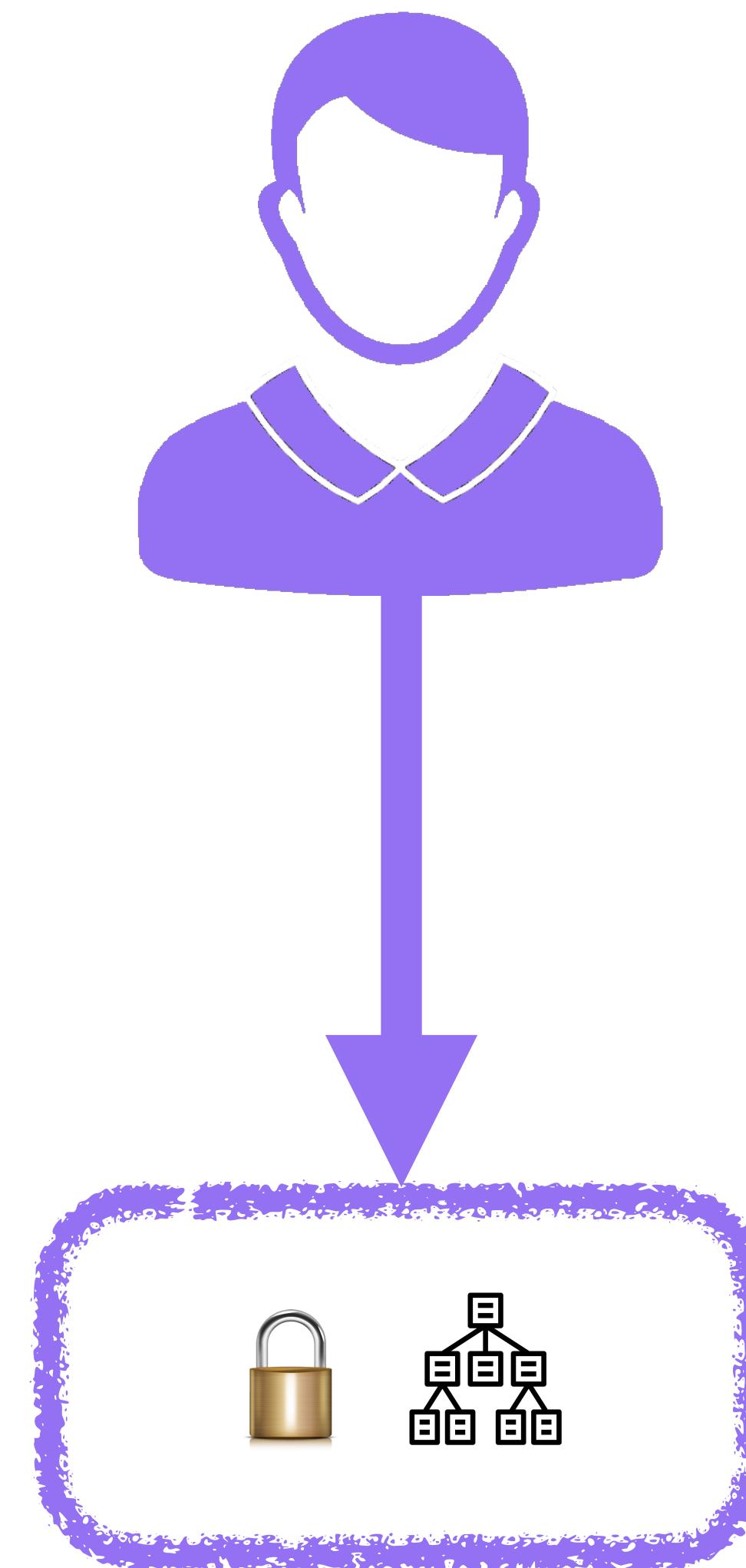
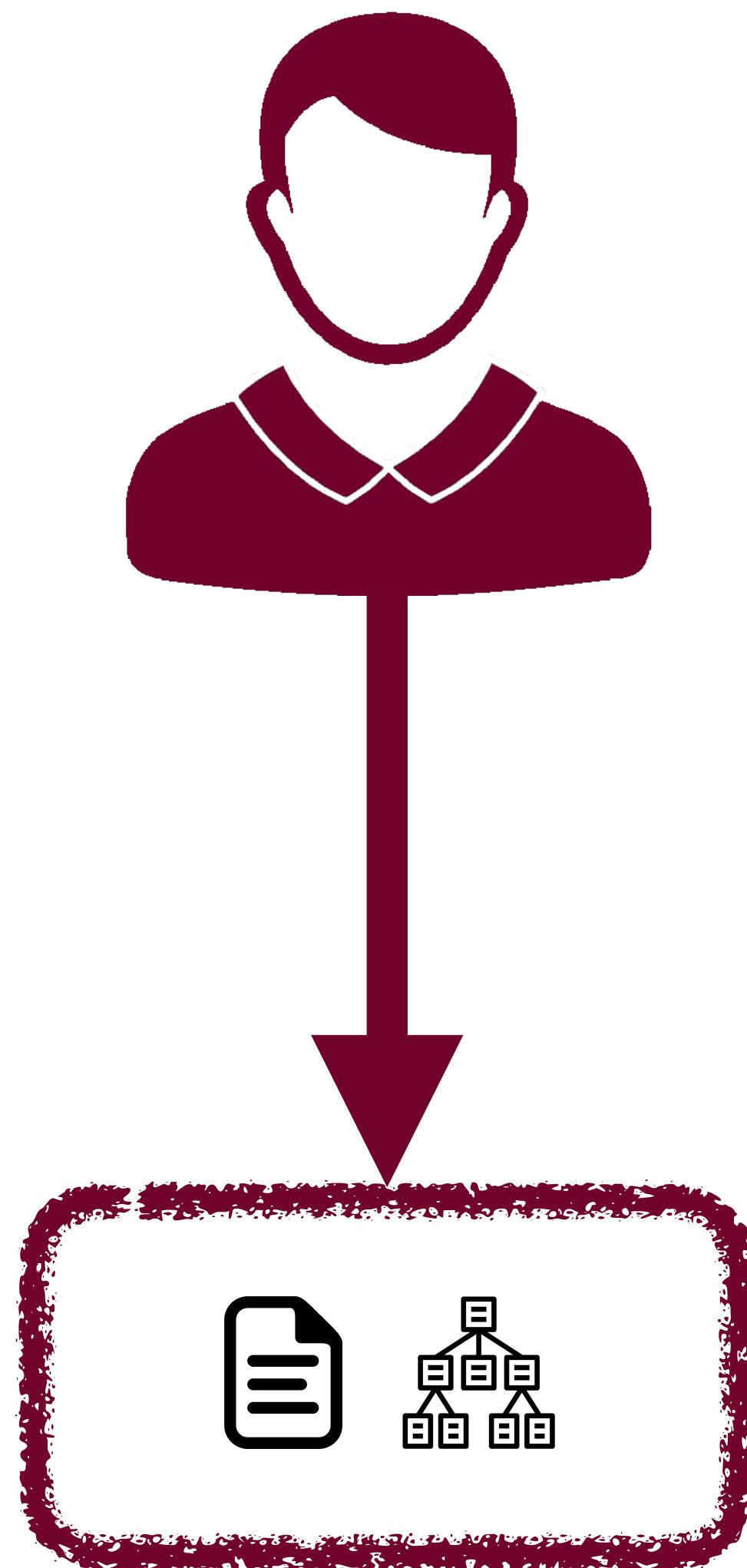


Logging

Core functionality

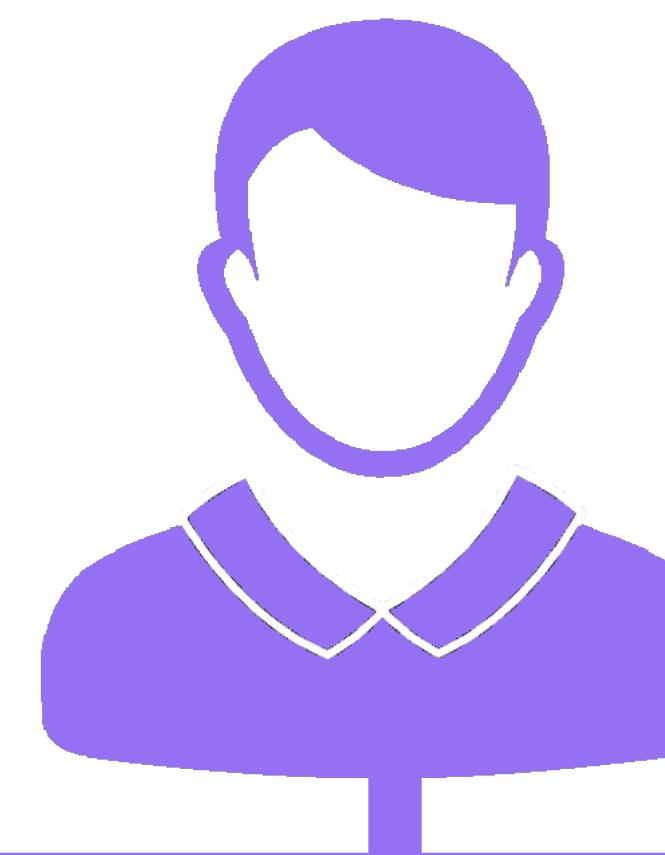
Design decisions affect functionality and quality attributes

Different Users Have Different Needs





Configuration:

A white rectangular box containing the word "Configuration:" in a dark red font. To the right of the text is a small icon consisting of a document symbol above a hierarchical tree structure of four levels.

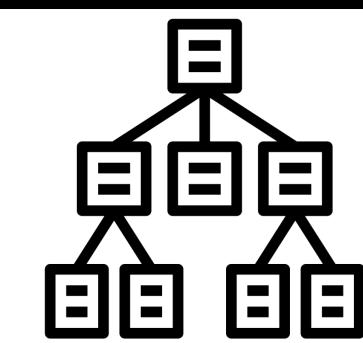
Configuration:

A white rectangular box containing the word "Configuration:" in a light purple font. To the right of the text is a small icon consisting of a padlock symbol above a hierarchical tree structure of four levels.

Configuration:

A white rectangular box containing the word "Configuration:" in a teal font. To the right of the text is a small icon consisting of a document symbol above a brown cardboard box.

Configurable System



Indexing



Encryption



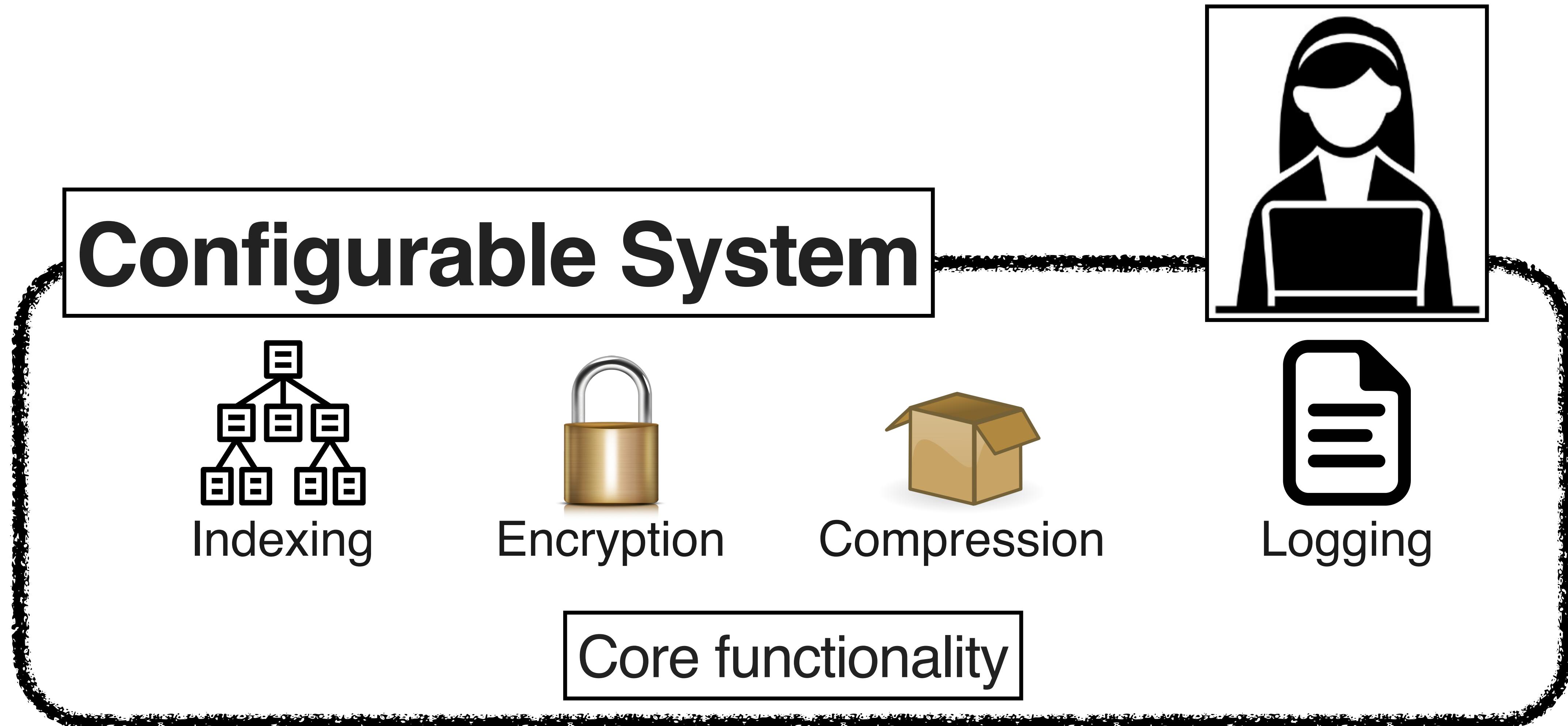
Compression



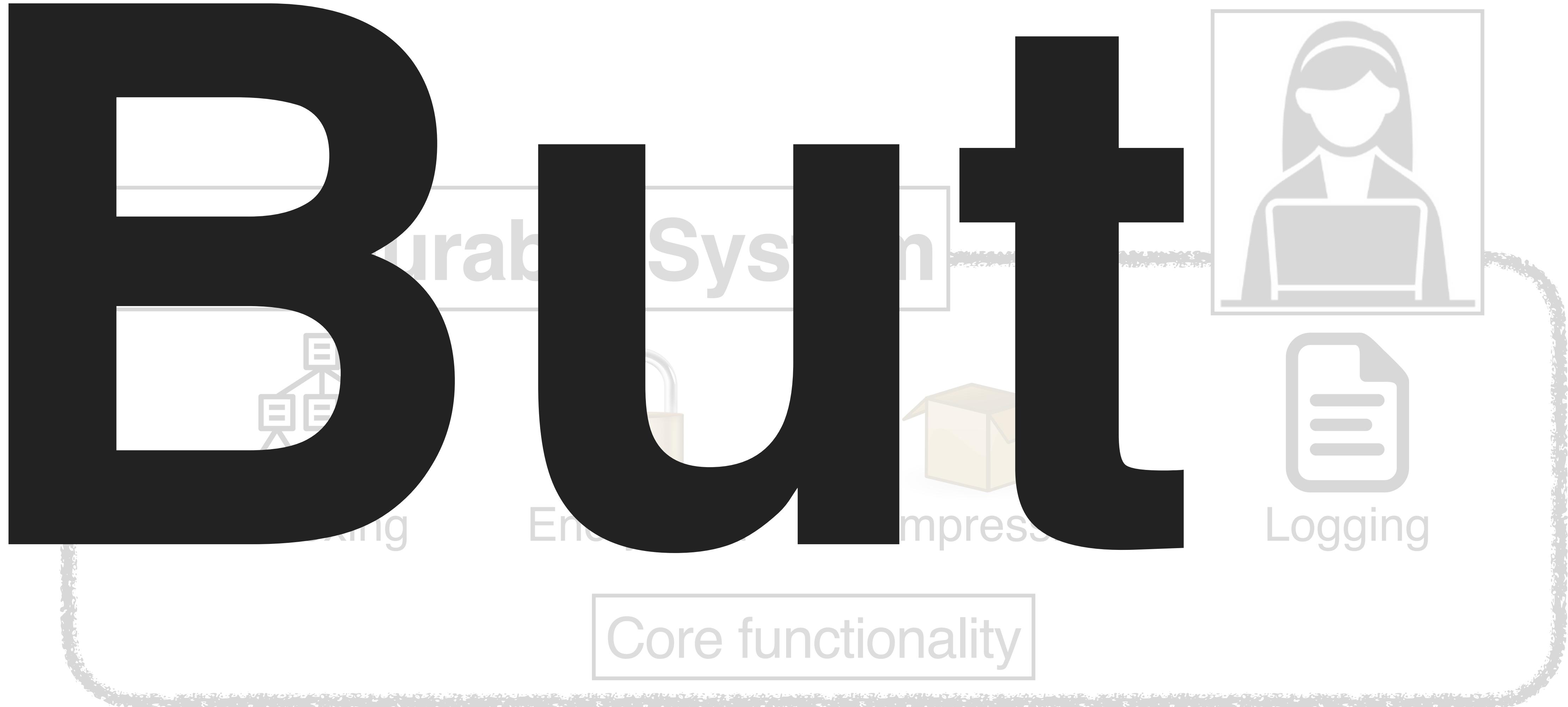
Logging

Core functionality

Developers Implement a Single System



Developers Implement a Single System





[Home](#) [Download](#) [Documentation](#) [Community](#) [Blog](#)

Manage massive amounts of data,
fast, without losing sleep

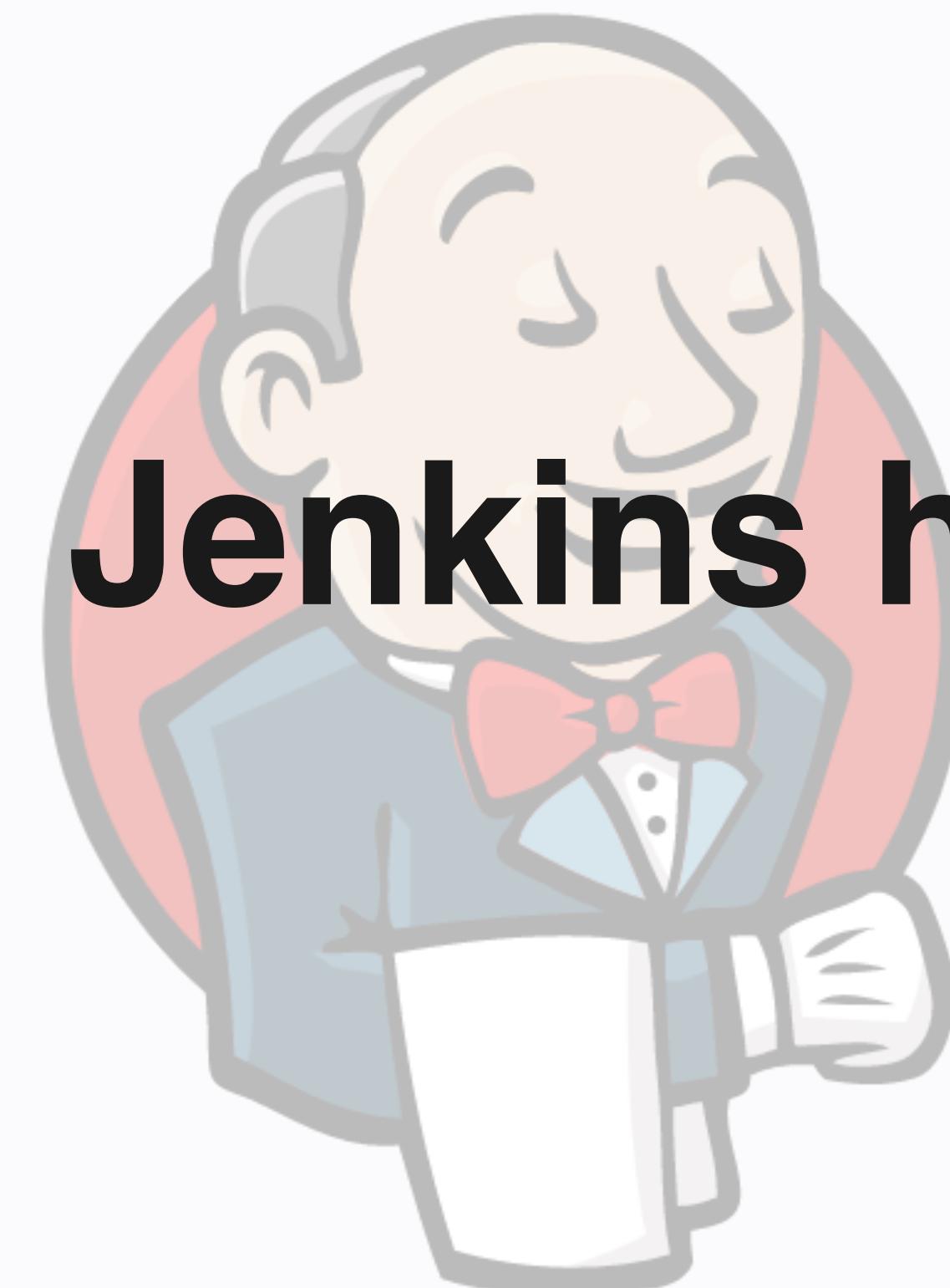
Cassandra has 170 options

[Download Cassandra](#)

Cassandra 4.0-beta3 Changelog

What is Cassandra?

The Apache Cassandra database is the right choice when you need scalability and high availability without compromising performance. [Linear scalability](#) and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra's support for replicating across multiple datacenters is best-in-class, providing lower latency for your users and the peace of mind of knowing that you can survive



Jenkins

Build great things at any scale

Jenkins has 186 options

The leading open source automation server. Jenkins provides hundreds of plugins to support building, deploying and automating any project.

[Documentation](#)

[Download](#)

APACHE KAFKA

Kafka's Broker has 208 options

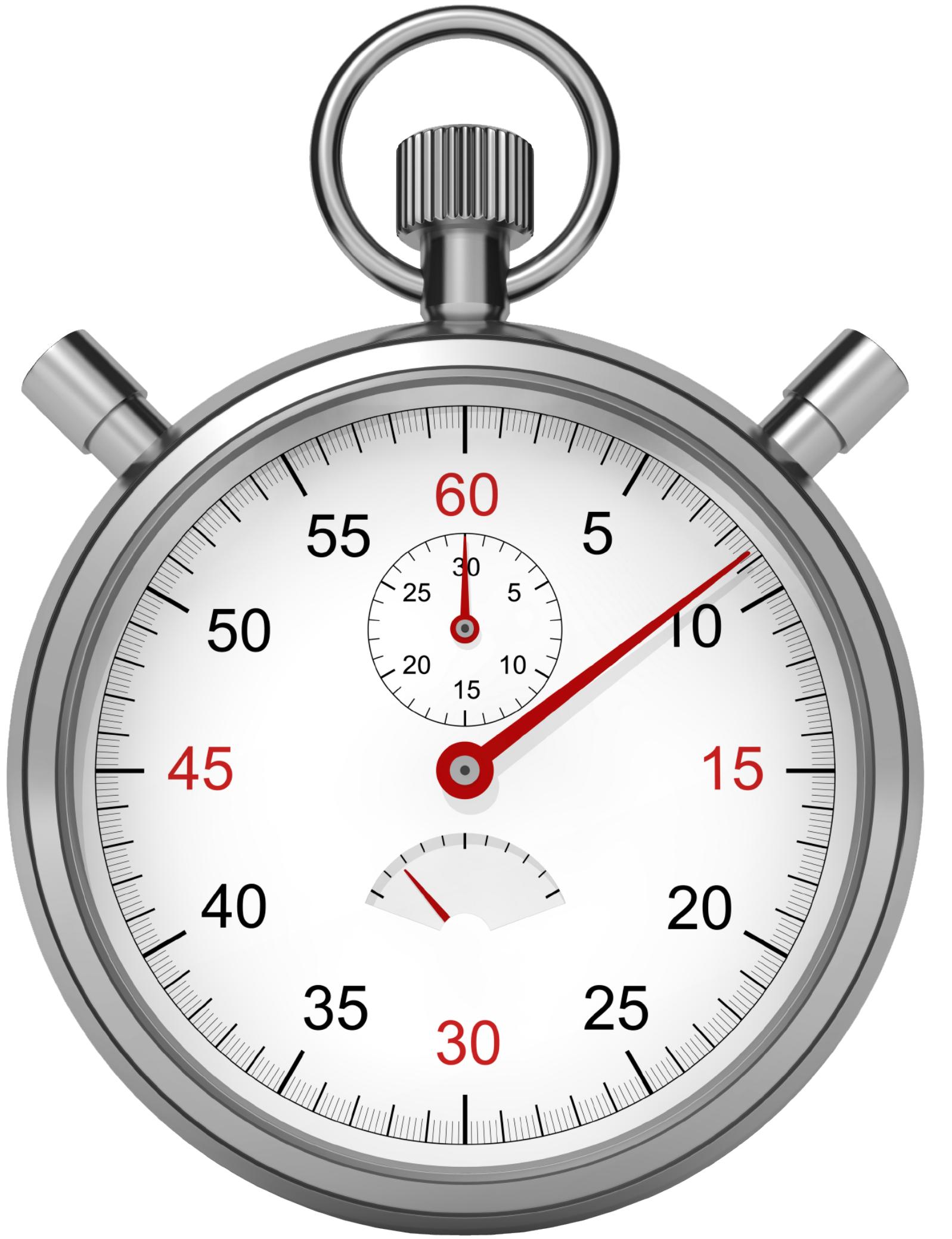
Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.

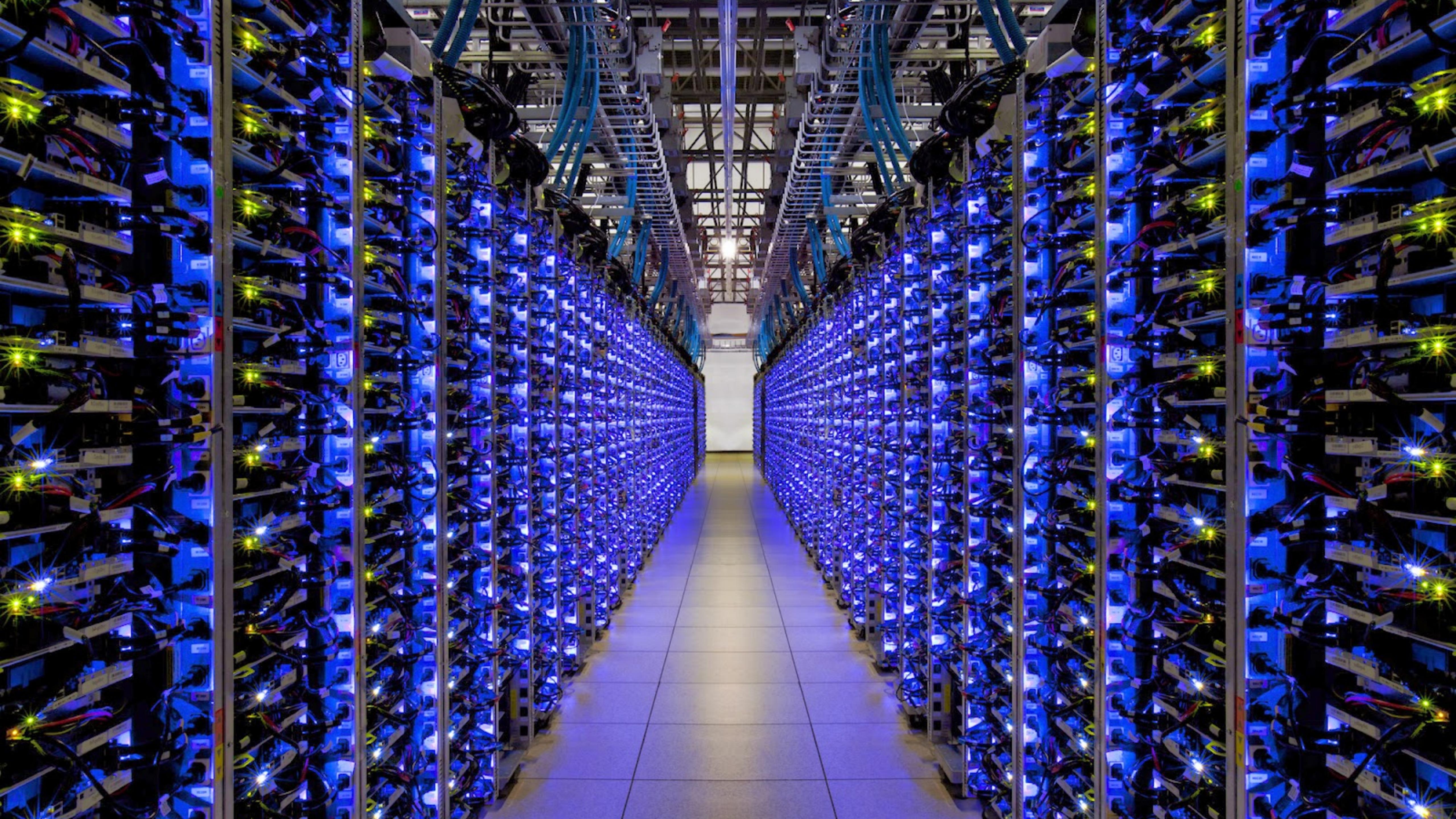


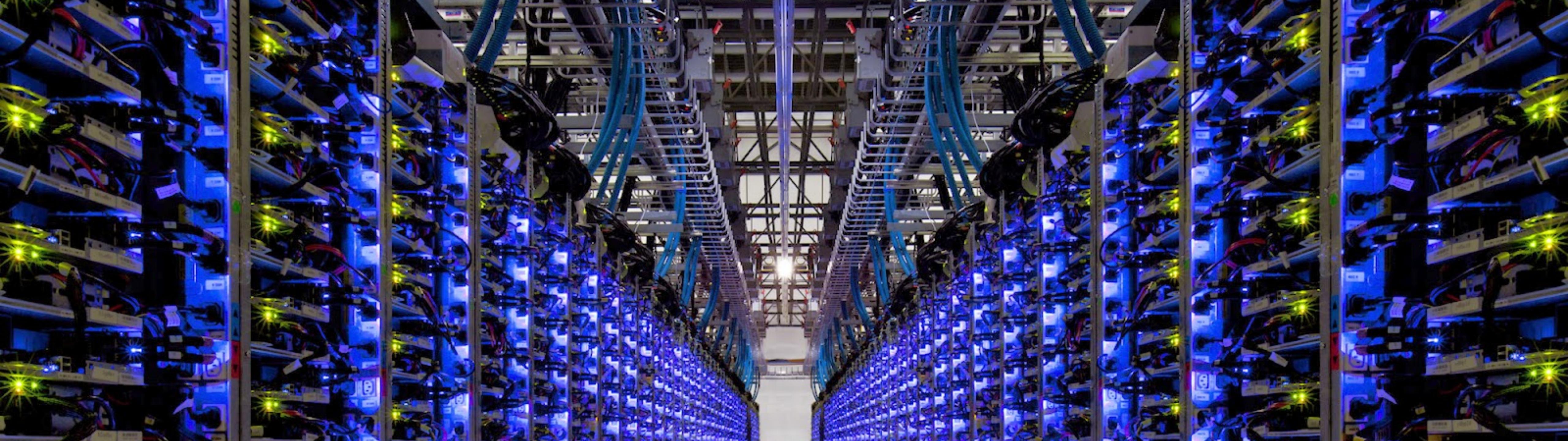
```
97 logviewer.port: 8000
98 logviewer.childopts: "-Xmx128m"
99 logviewer.cleanup.age.mins: 10080
100 logviewer.appendер.name: "A1"
101 logviewer.max.sum.worker.logs.size.mb: 4096
102 logviewer.max.per.worker.logs.size.mb: 2048
103
104 logs.users: null
105
106 drpc.port: 3772
107 drpc.worker.threads: 64
108 drpc.max_buffer_size: 1048576
109 drpc.queue.size: 128
110 drpc.invocations.port: 3773
111 drpc.invocations.threads: 64
112 drpc.request.timeout.secs: 600
113 drpc.childopts: "-Xmx768m"
114 drpc.http.port: 3774
115 drpc.https.port: -1
116 drpc.https.keystore.password: ""
117 drpc.https.keystore.type: "JKS"
118 drpc.http.creds.plugin: org.apache.storm.security.auth.DefaultHttpCredentialsPlugin
119 drpc.authorizer.acl.filename: "drpc-auth-acl.yaml"
120 drpc.authorizer.acl.strict: false
121
122 transactional.zookeeper.root: "/transactional"
123 transactional.zookeeper.servers: null
124 transactional.zookeeper.port: null
125
126 ## blobstore configs
127 supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
128 supervisor.blobstore.download.thread.count: 5
129 supervisor.blobstore.download.max_retries: 3
130 supervisor.localizer.cache.target.size.mb: 10240
131 supervisor.localizer.cleanup.interval.ms: 600000
132
133 nimbus.blobstore.class: "org.apache.storm.blobstore.LocalFsBlobStore"
134 nimbus.blobstore.expiration.secs: 600
135
136 storm.blobstore.inputstream.buffer.size.bytes: 65536
137 client.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
138 storm.blobstore.replication.factor: 3
```

Challenging to understand how options affect functionality and quality attributes in exponentially large configuration spaces

```
97 logviewer.port: 8000
98 logviewer.childopts: "-Xmx128m"
99 logviewer.cleanup.age.mins: 10080
100 logviewer.appendер.name: "A1"
101 logviewer.max.sum.worker.logs.size.mb: 4096
102 logviewer.max.per.worker.logs.size.mb: 2048
103
104 logs.users: null
105
106 drpc.port: 3772
107 drpc.worker.threads: 64
108 drpc.max_buffer_size: 1048576
109 drpc.queue.size: 128
110 drpc.invocations.port: 3773
111 drpc.invocations.threads: 64
112 drpc.request.timeout.secs: 600
113 drpc.request.timeout.millis: 600000
114 drpc.https.port: 8774
115 drpc.https.port: -1
116 drpc.https.keystore.password: ""
117 drpc.https.keystore.type: "JKS"
118 drpc.https.keystore.path: "/etc/storm/ssl/storm-https.jks"
119 drpc.https.keystore.alias: "storm-https"
120 drpc.authorizer.acl.strict: false
121
122 transactional.zookeeper.port: null
123 transactional.zookeeper.addresses: null
124 transactional.zookeeper.port: null
125
126 ## blobstore configs
127 supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
128 supervisor.blobstore.download.thread.count: 5
129 supervisor.blobstore.download.max_retries: 3
130 supervisor.localizer.cache.target.size.mb: 10240
131 supervisor.localizer.cleanup.interval.ms: 600000
132
133 nimbus.blobstore.class: "org.apache.storm.blobstore.LocalFsBlobStore"
134 nimbus.blobstore.expiration.secs: 600
135
136 storm.blobstore.inputstream.buffer.size.bytes: 65536
137 client.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
138 storm.blobstore.replication.factor: 3
```







Performance

Energy consumption

Execution time

Operational costs



User

- Make tradeoff decisions
- Answer what-if questions
- Run system efficiently
- Satisfy their needs and requirements



Developer

- Design, implement, and maintain efficient software
- Debug surprising performance behavior

Configuration space problem

- Answer what-if questions
- Run system efficiently
- Satisfy their needs and requirements

Describe performance in exponentially large configuration spaces



Developer

- Design, implement, and maintain efficient software
- Debug surprising behavior



Optimization!



Optimization!

Reduces:

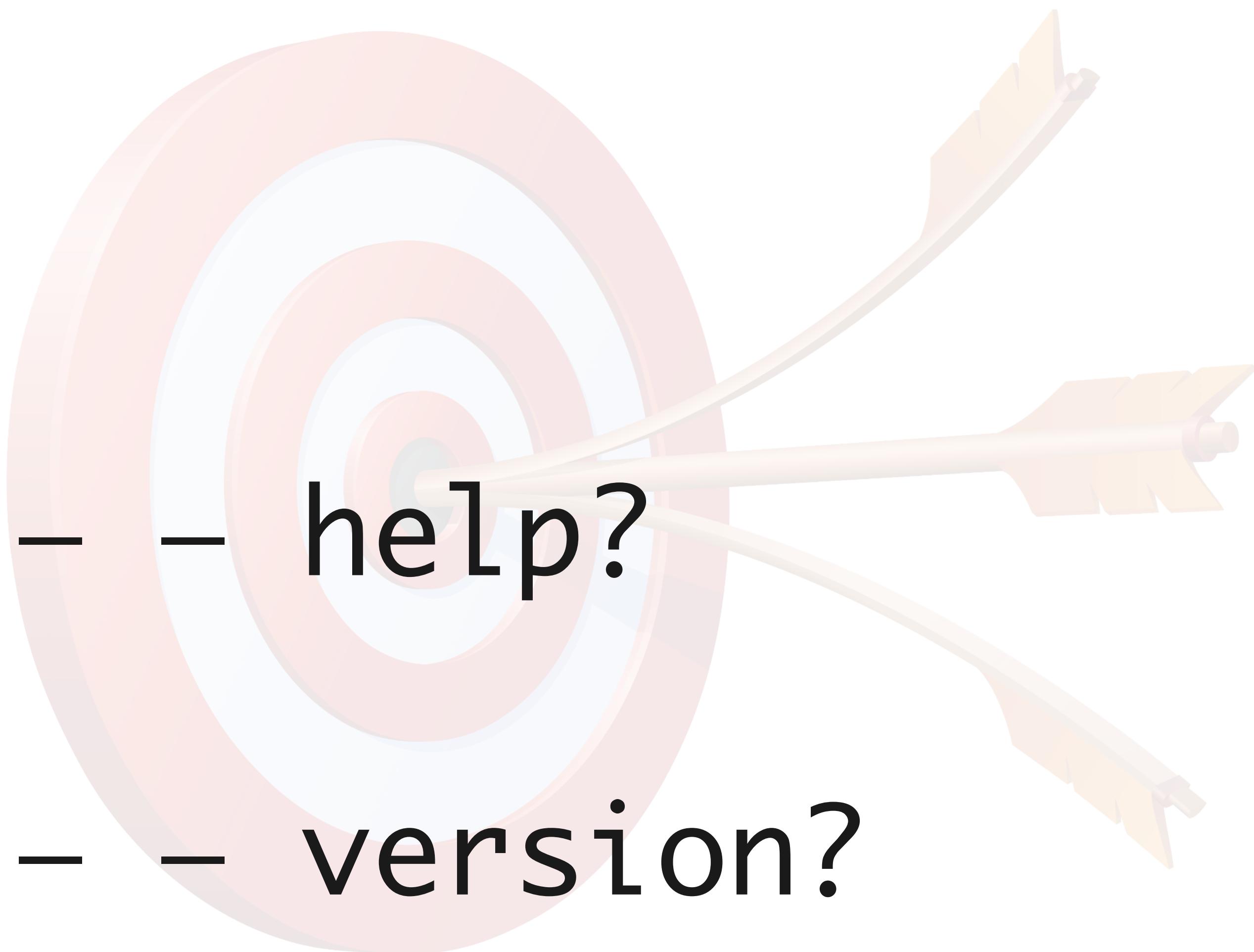
- Execution time**
- Energy consumption**
- Operational cost**



Optimization!

— help?

— version?



Reduces:

Execution time

Memory consumption

Operation cost





User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

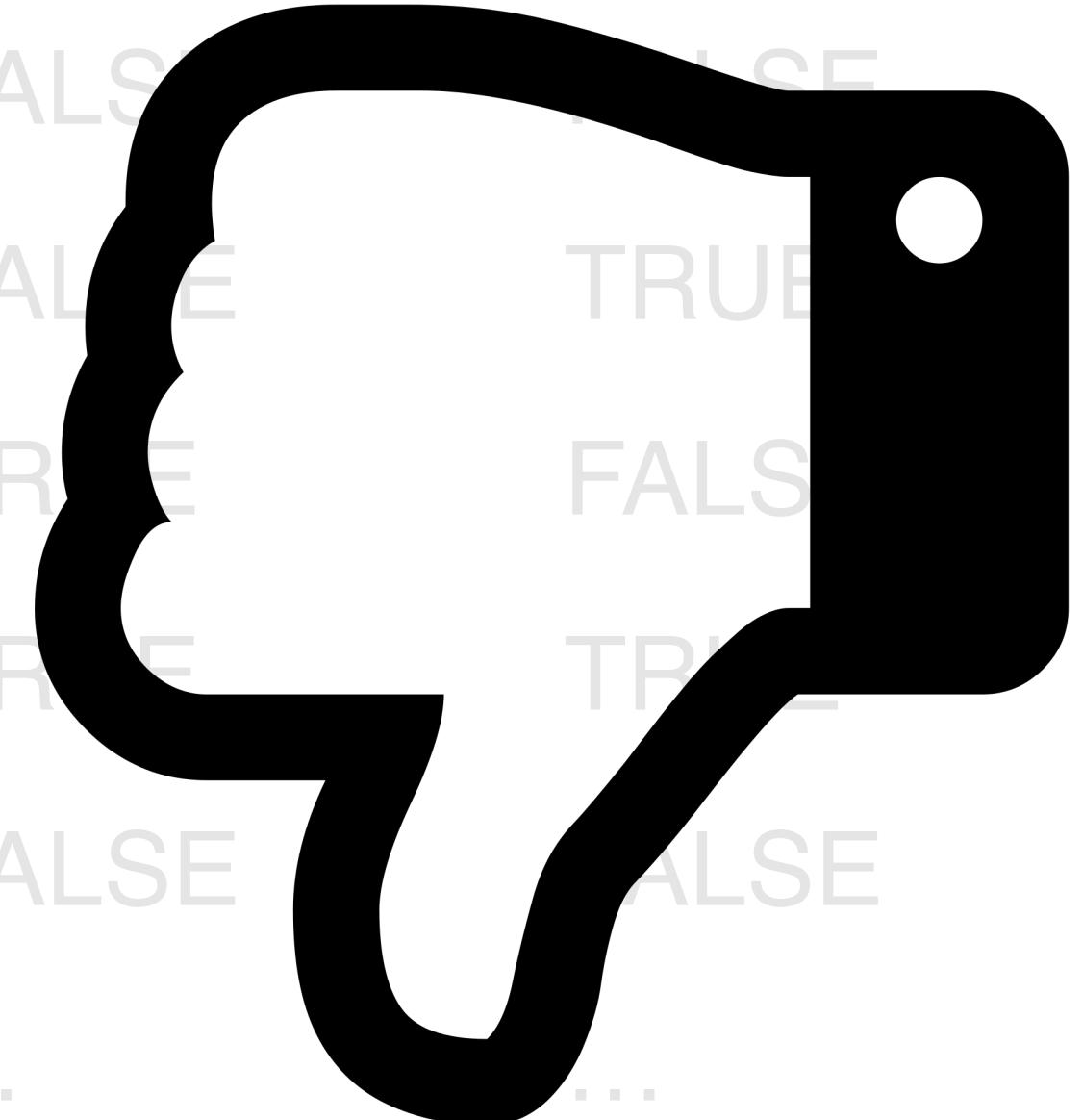
Brute-force Approach

Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	25
TRUE	FALSE	FALSE	TRUE	...	34
TRUE	FALSE	TRUE	FALSE	...	28
TRUE	FALSE	TRUE	TRUE	...	37
TRUE	TRUE	FALSE	FALSE	...	20
TRUE	TRUE	FALSE	TRUE	...	29
TRUE	TRUE	TRUE	FALSE	...	23
TRUE	TRUE	TRUE	TRUE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...

Brute-force Approach

Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	25
TRUE	FALSE	FALSE	TRUE	...	34
TRUE	FALSE	TRUE	TRUE	...	37
TRUE	TRUE	FALSE	TRUE	...	20
TRUE	TRUE	FALSE	TRUE	...	29
TRUE	TRUE	TRUE	FALSE	...	23
TRUE	TRUE	TRUE	TRUE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...	23

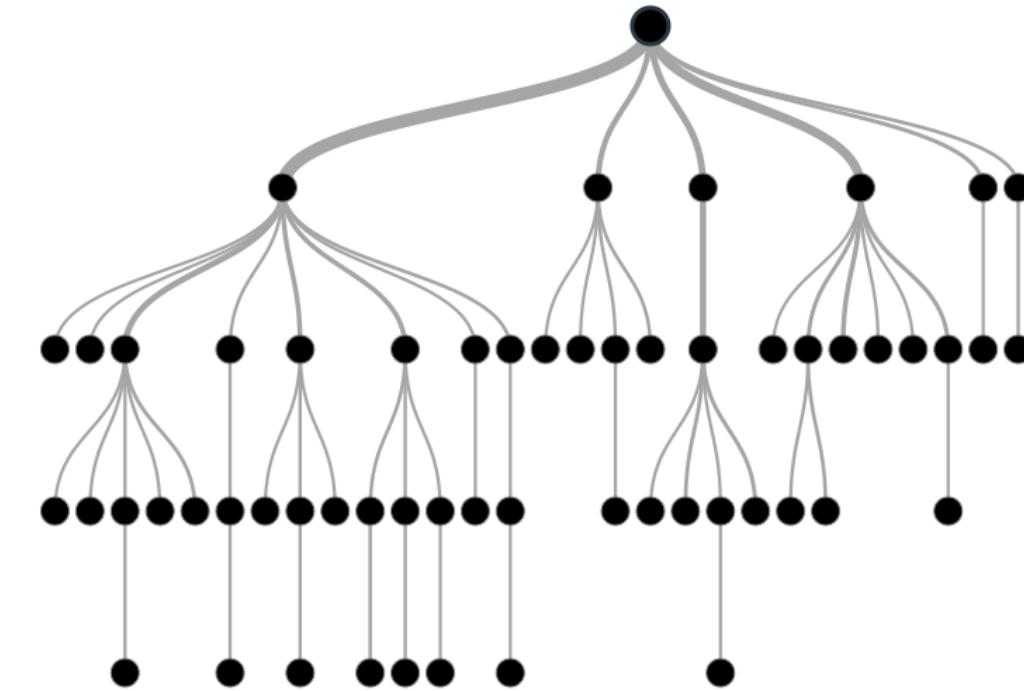
Exponentially large configuration spaces



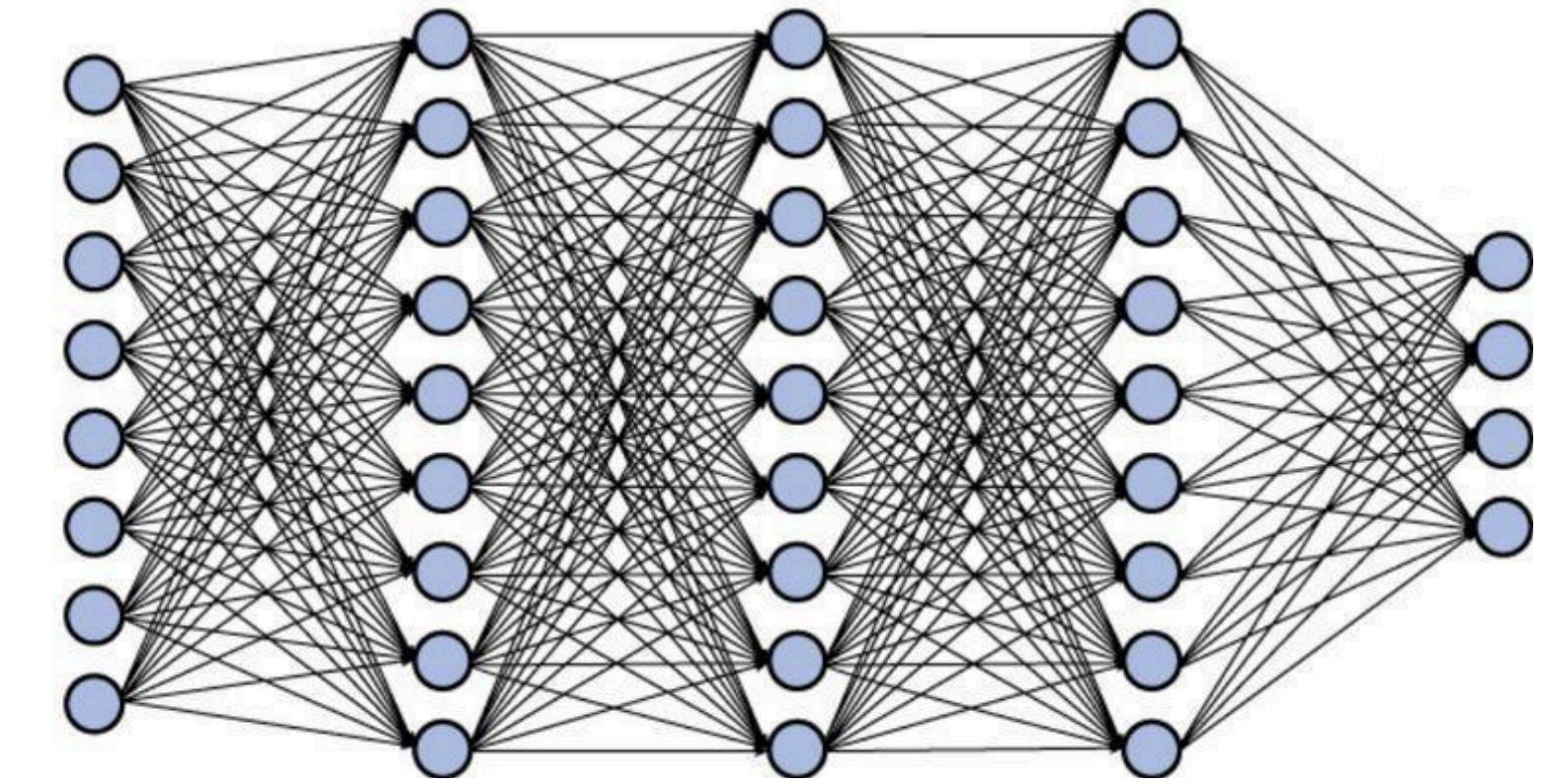
Performance-Influence Models

$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption

Sparse linear models



Decision trees/
Random forests



Neural networks

Performance-Influence Models

T = 25

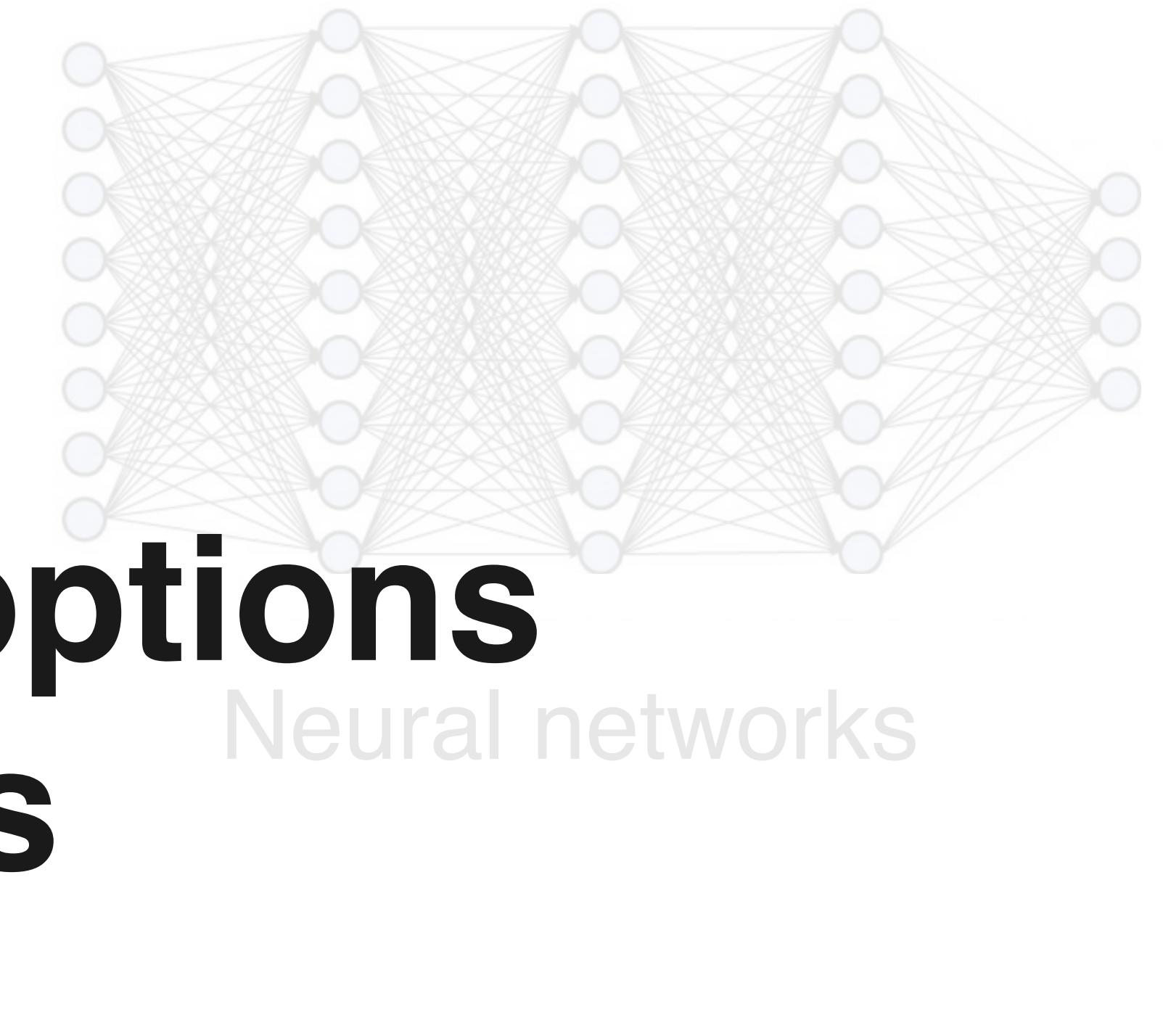
Useful for:

Making predictions

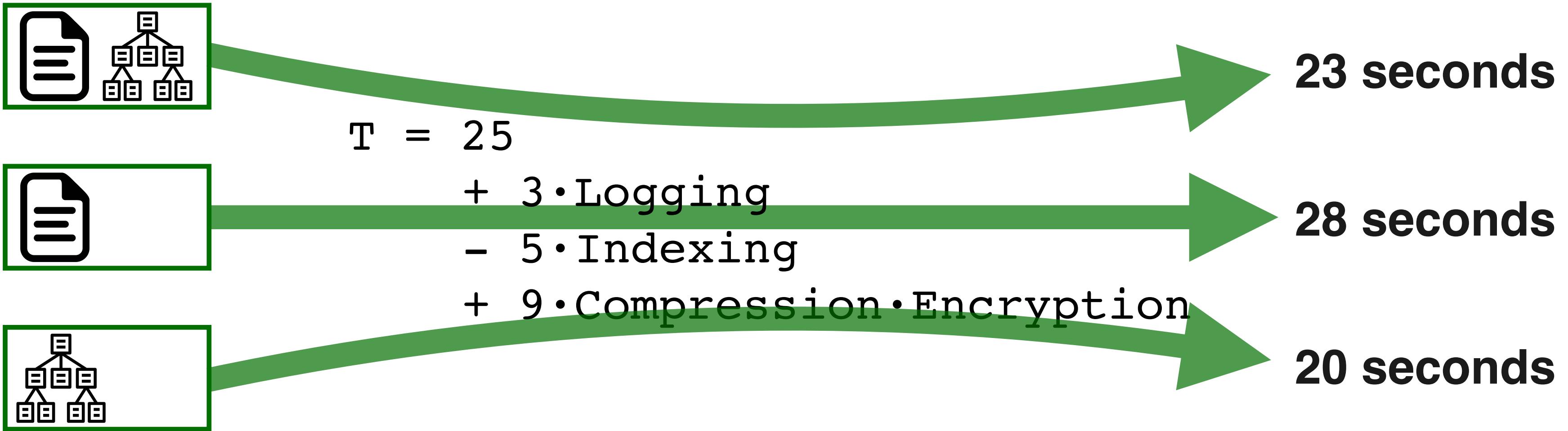
Understanding effect of options

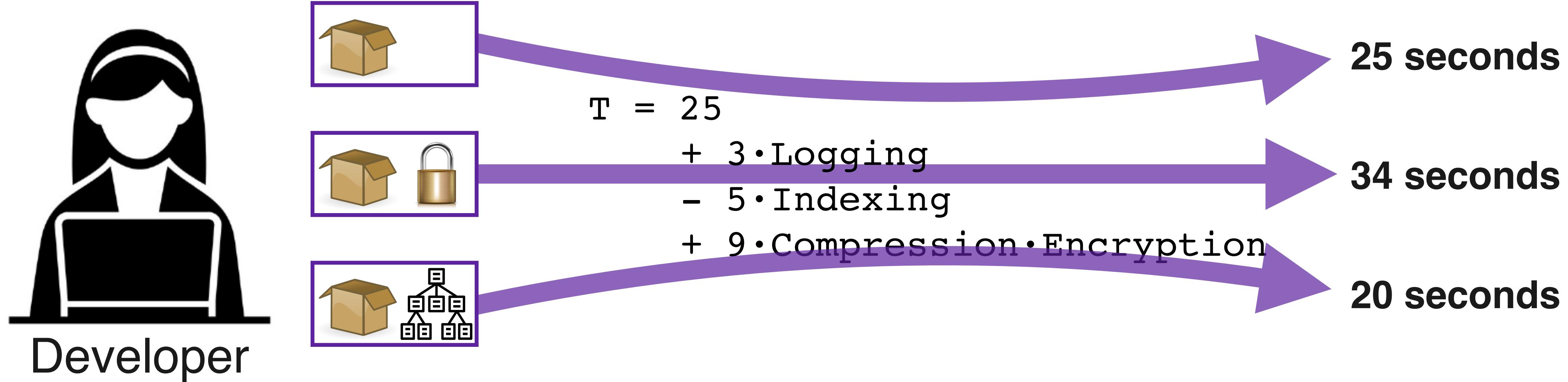
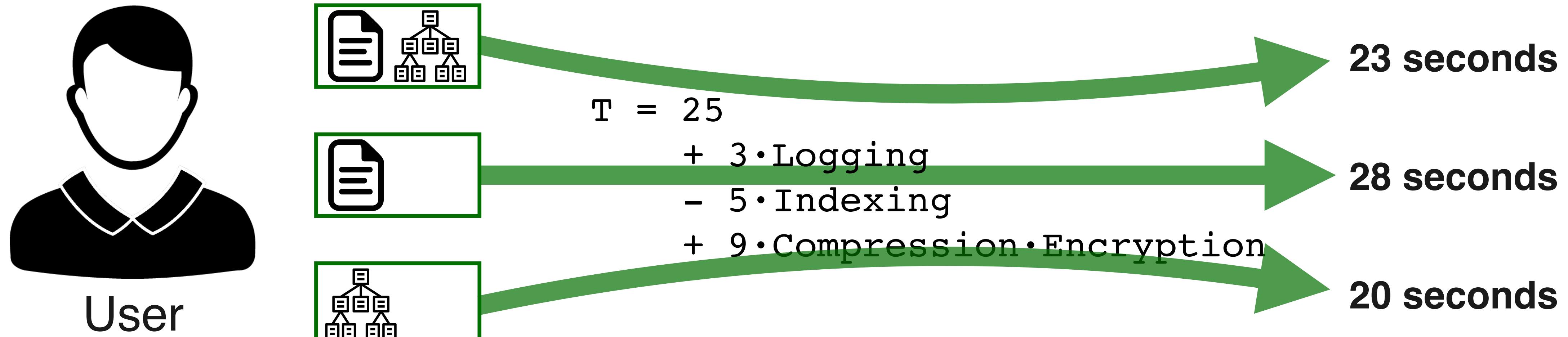
Making tradeoff decisions

Decision trees/
Random forests



Neural networks





Black-box Approaches

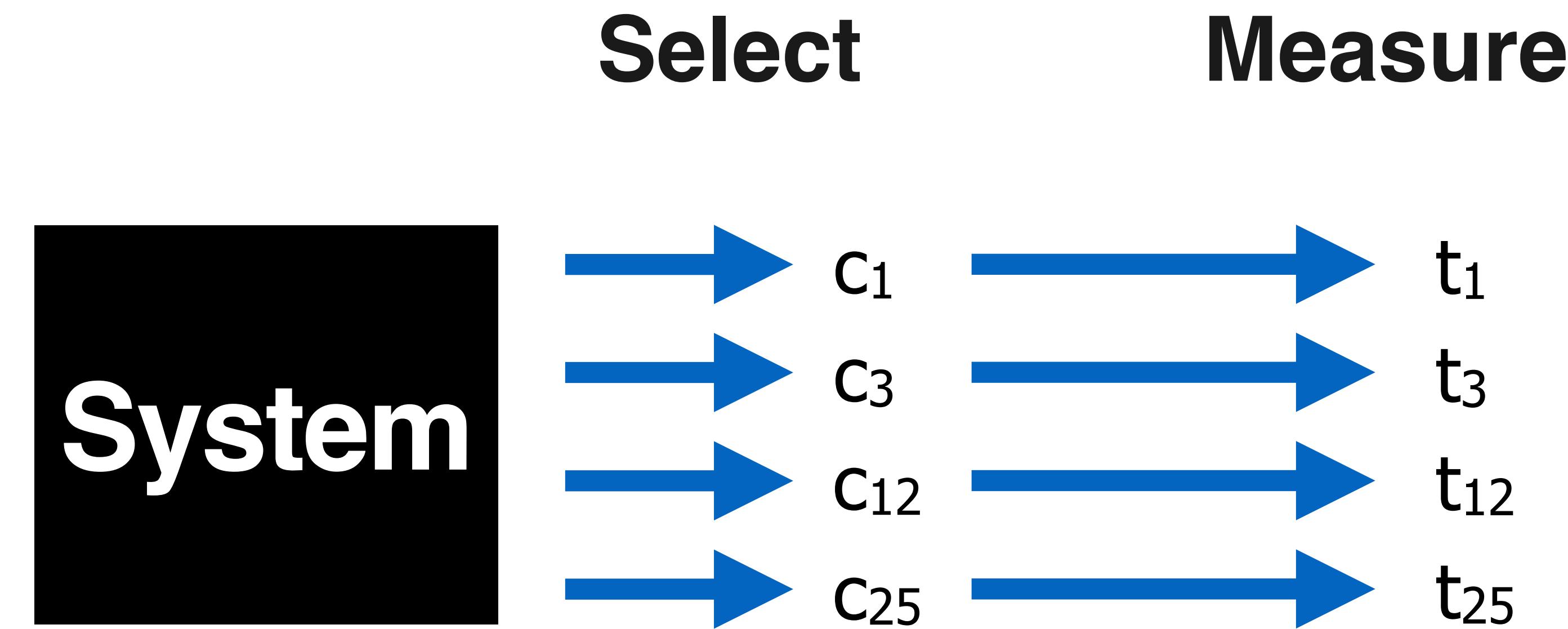
System

Black-box Approaches

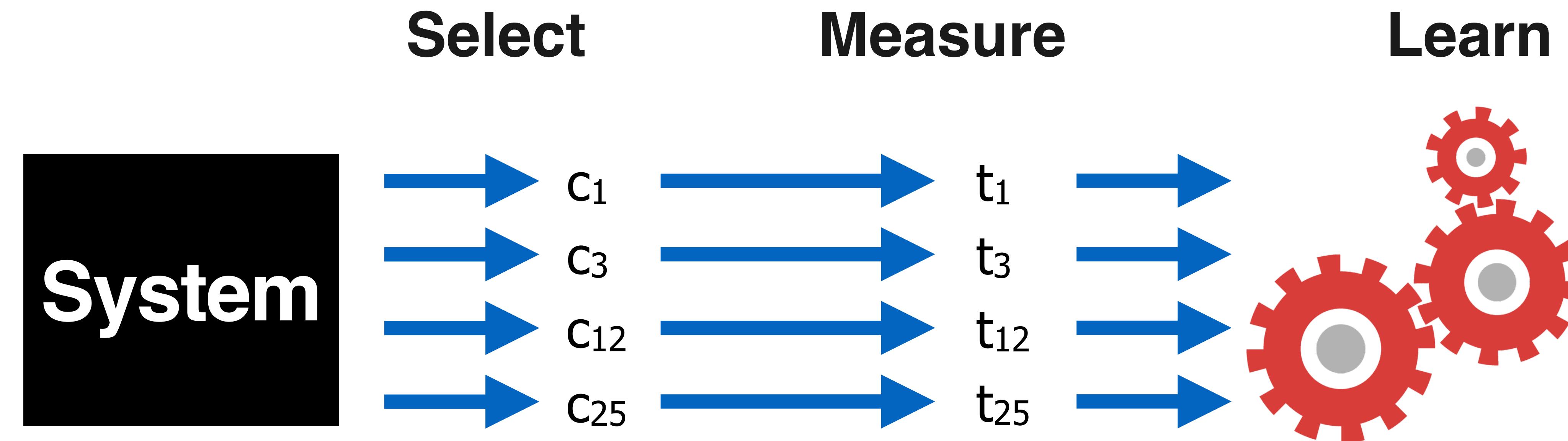
Select



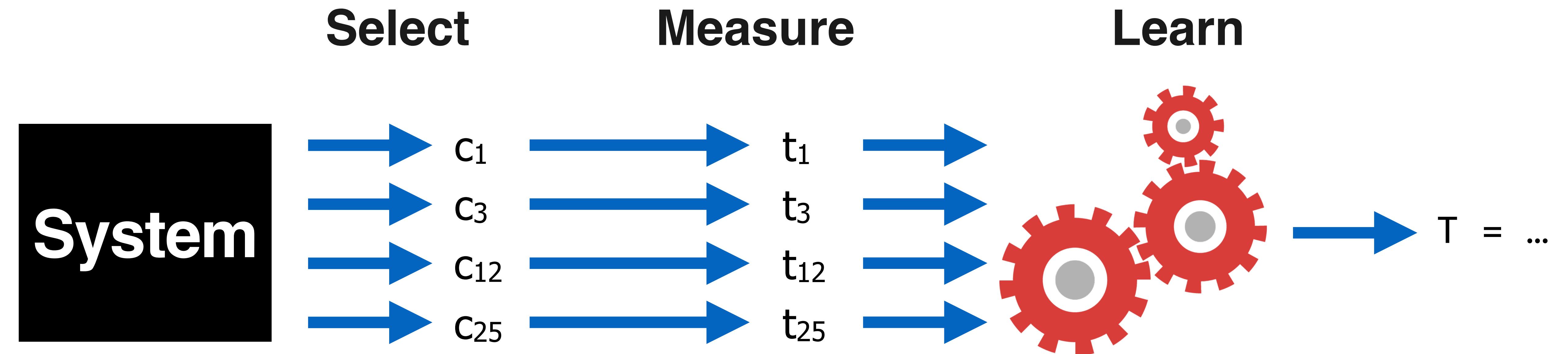
Black-box Approaches



Black-box Approaches



Black-box Approaches



$$\begin{aligned}
 T = & 25 \\
 & + 3 \cdot \text{Logging} \\
 & - 5 \cdot \text{Indexing} \\
 & + 9 \cdot \text{Compression} \cdot \text{Encryption}
 \end{aligned}$$

Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	20
TRUE	FALSE	FALSE	TRUE	...	23
TRUE	FALSE	TRUE	FALSE	...	20
TRUE	FALSE	TRUE	TRUE	...	23
TRUE	TRUE	FALSE	FALSE	...	20
TRUE	TRUE	FALSE	TRUE	...	23
TRUE	TRUE	TRUE	FALSE	...	29
TRUE	TRUE	TRUE	TRUE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...

$$\begin{aligned}
 T = & 25 \\
 & + 3 \cdot \text{Logging} \\
 & - 5 \cdot \text{Indexing} \\
 & + 9 \cdot \text{Compression} \cdot \text{Encryption}
 \end{aligned}$$

Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	20
TRUE	FALSE	FALSE	TRUE	...	23
TRUE	FALSE	TRUE	FALSE	...	20
TRUE	FALSE	TRUE	TRUE	...	23
TRUE	TRUE	FALSE	FALSE	...	20
TRUE	TRUE	FALSE	TRUE	...	23
TRUE	TRUE	TRUE	FALSE	...	29
TRUE	TRUE	TRUE	TRUE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...

$$T = 25 + 3 \cdot \text{Logging} - 5 \cdot \text{Indexing} + 9 \cdot \text{Compression} \cdot \text{Encryption}$$

Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	20
TRUE	FALSE	FALSE	TRUE	...	23
TRUE	FALSE	TRUE	TRUE	...	23
TRUE	TRUE	FALSE	FALSE	...	20
TRUE	TRUE	FALSE	TRUE	...	23
TRUE	TRUE	TRUE	FALSE	...	29
TRUE	TRUE	TRUE	TRUE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...

Might explore irrelevant interactions

$$\begin{aligned}
 T = & 25 \\
 & + 3 \cdot \text{Logging} \\
 & - 5 \cdot \text{Indexing} \\
 & + 9 \cdot \text{Compression} \cdot \text{Encryption}
 \end{aligned}$$

Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	20
TRUE	FALSE	FALSE	TRUE	...	23
TRUE	FALSE	TRUE	FALSE	...	20
TRUE	FALSE	TRUE	TRUE	...	23
TRUE	TRUE	FALSE	FALSE	...	20
TRUE	TRUE	FALSE	TRUE	...	23
TRUE	TRUE	TRUE	FALSE	...	29
TRUE	TRUE	TRUE	TRUE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...

$$\begin{aligned}
 T = & 25 \\
 & + 3 \cdot \text{Logging} \\
 & - 5 \cdot \text{Indexing} \\
 & + 9 \cdot \text{Compression} \cdot \text{Encryption}
 \end{aligned}$$

Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	20
TRUE	FALSE	FALSE	TRUE	...	23
TRUE	FALSE	TRUE	TRUE	...	20
TRUE	TRUE	FALSE	FALSE	...	23
TRUE	TRUE	FALSE	TRUE	...	20
TRUE	TRUE	TRUE	FALSE	...	23
TRUE	TRUE	TRUE	TRUE	...	29
FALSE	FALSE	FALSE	FALSE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...

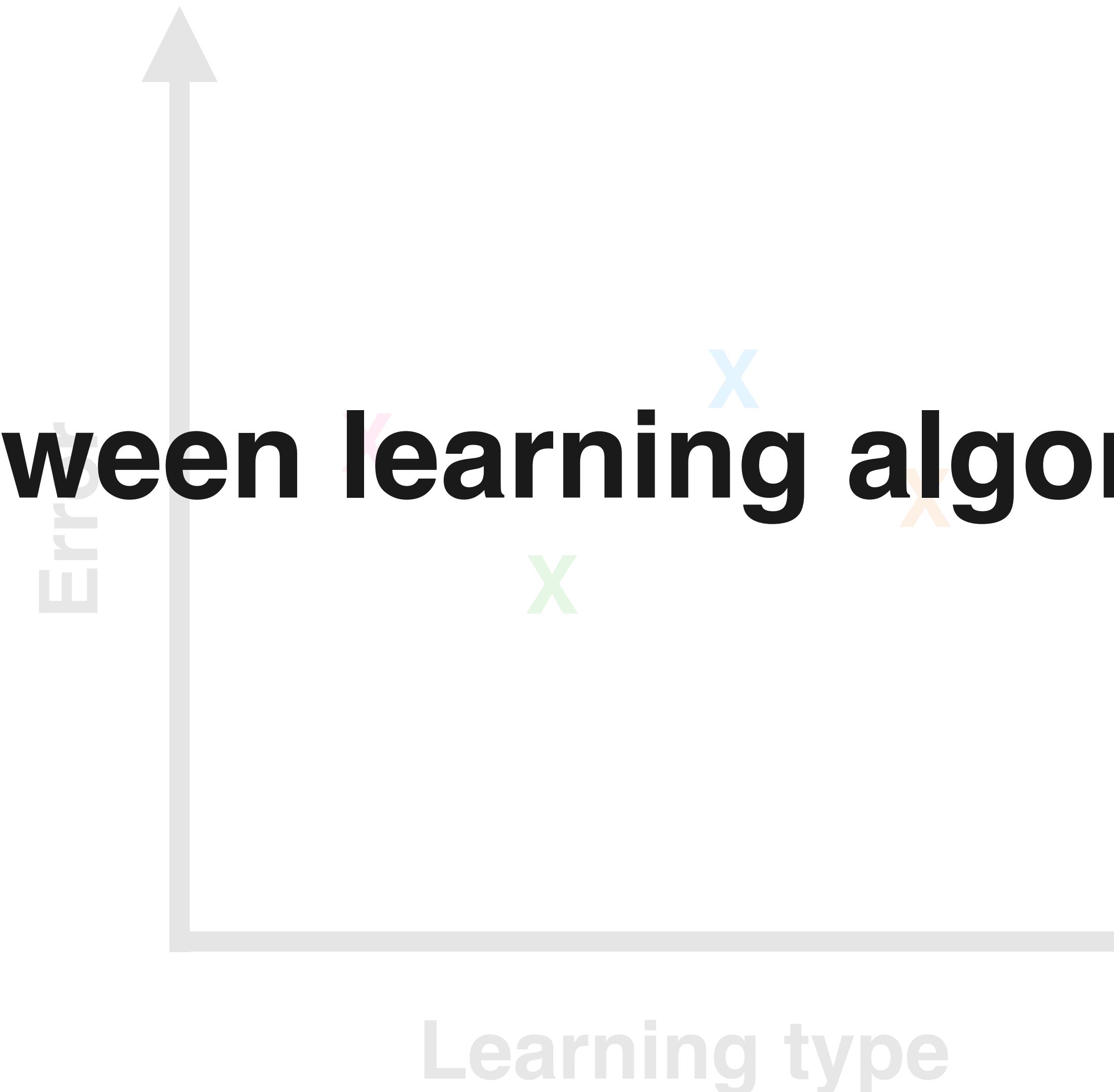
Might miss relevant performance interactions

$$\begin{aligned}
 T = & 25 \\
 & + 3 \cdot \text{Logging} \\
 & - 5 \cdot \text{Indexing} \\
 & + 9 \cdot \text{Compression} \cdot \text{Encryption}
 \end{aligned}$$

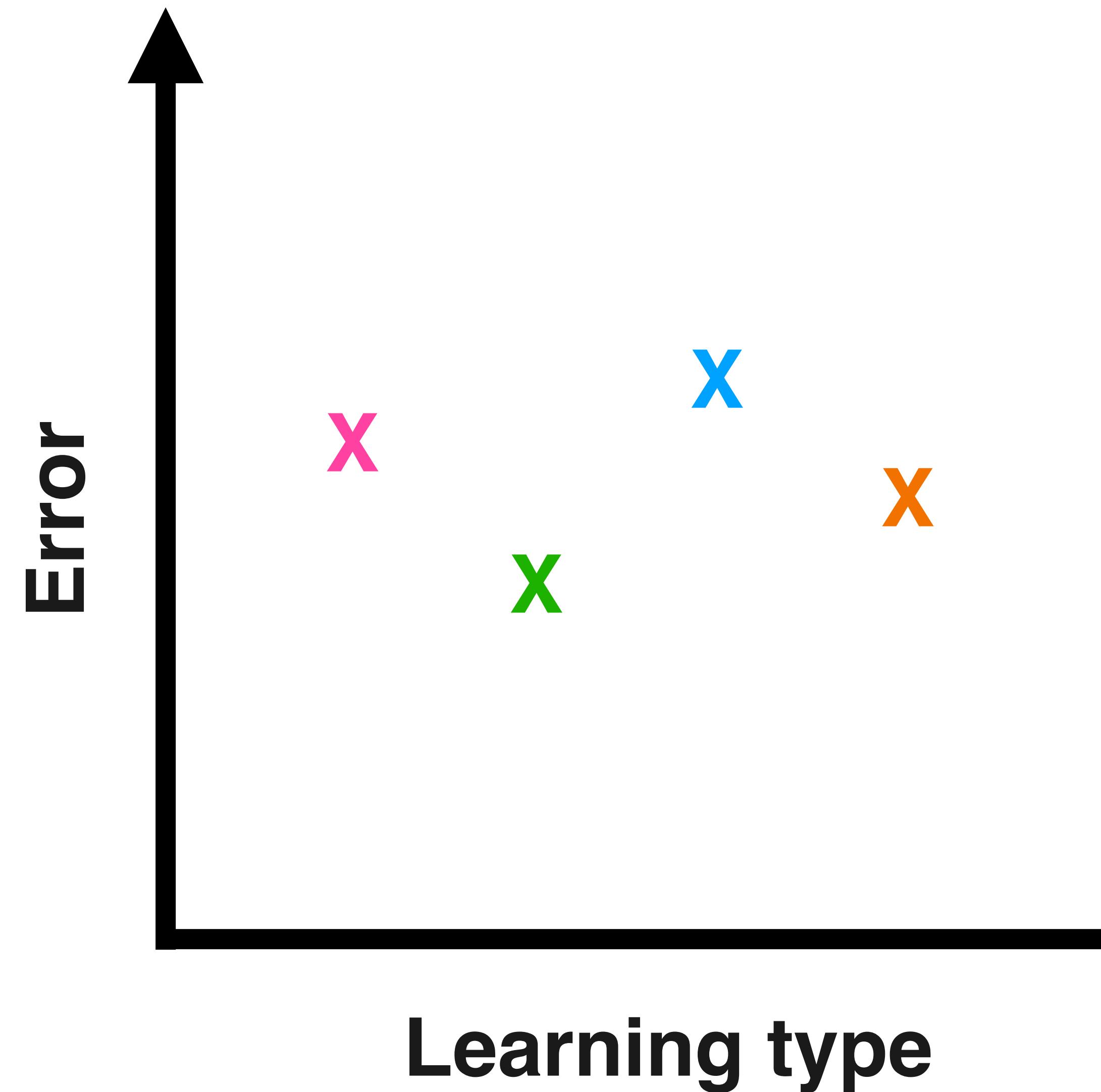
Indexing	Encryption	Compression	Logging	...	Time
TRUE	FALSE	FALSE	FALSE	...	20
TRUE	FALSE	FALSE	TRUE	...	23
TRUE	FALSE	TRUE	FALSE	...	20
TRUE	FALSE	TRUE	TRUE	...	23
TRUE	TRUE	FALSE	FALSE	...	20
TRUE	TRUE	FALSE	TRUE	...	23
TRUE	TRUE	TRUE	FALSE	...	29
TRUE	TRUE	TRUE	TRUE	...	32
FALSE	FALSE	FALSE	FALSE	...	25
...

*Equal number
of samples

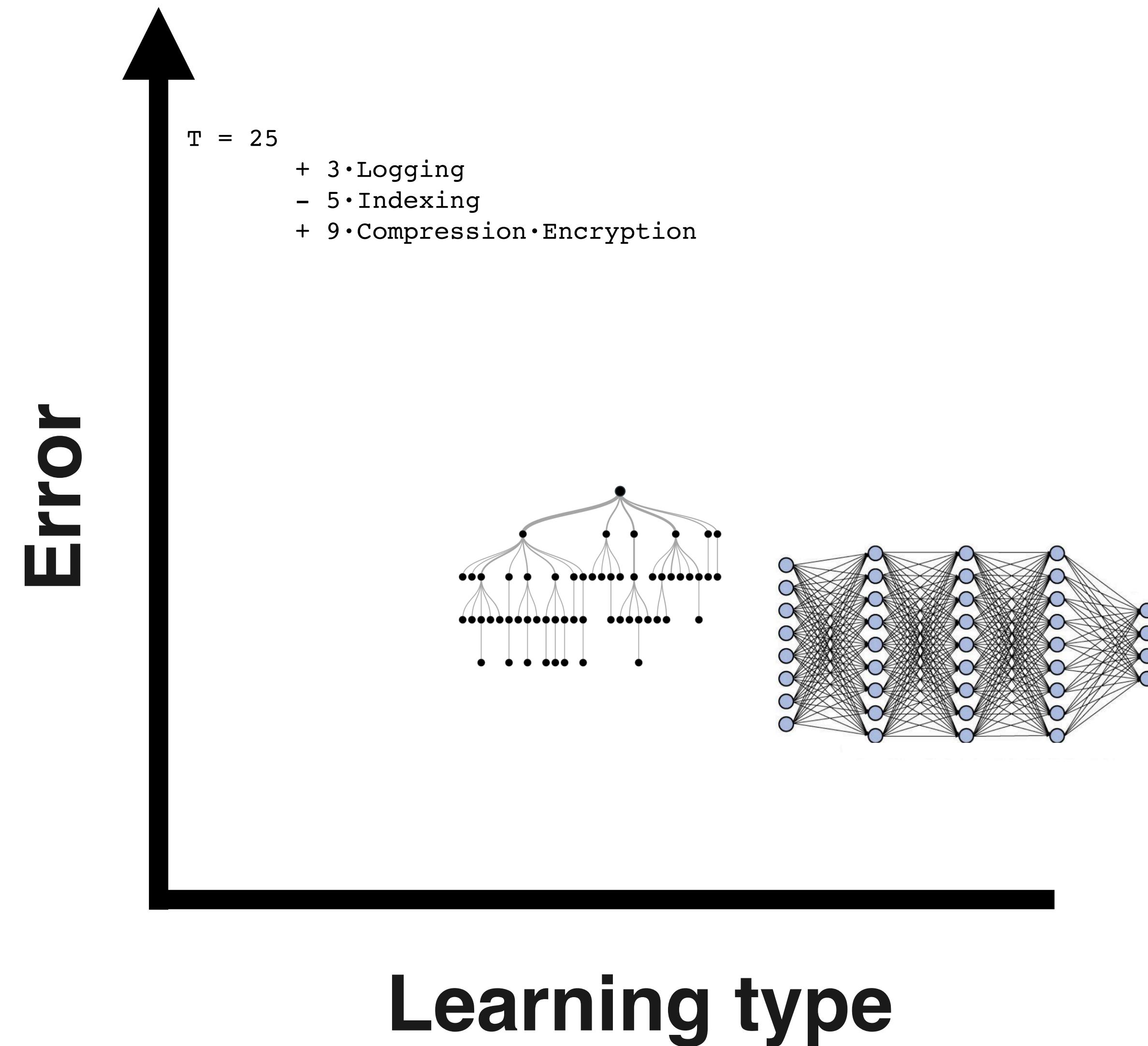
Tradeoff between learning algorithm and accuracy



***Equal number
of samples**



***Equal number
of samples**



*Equal number

Accurate predictions



*Equal number
of samples

Accurate predictions

+ 5 - 5 + 5
+ 5 - 5 + 5
+ 9 • Compression • Encryption

Understanding and debugging

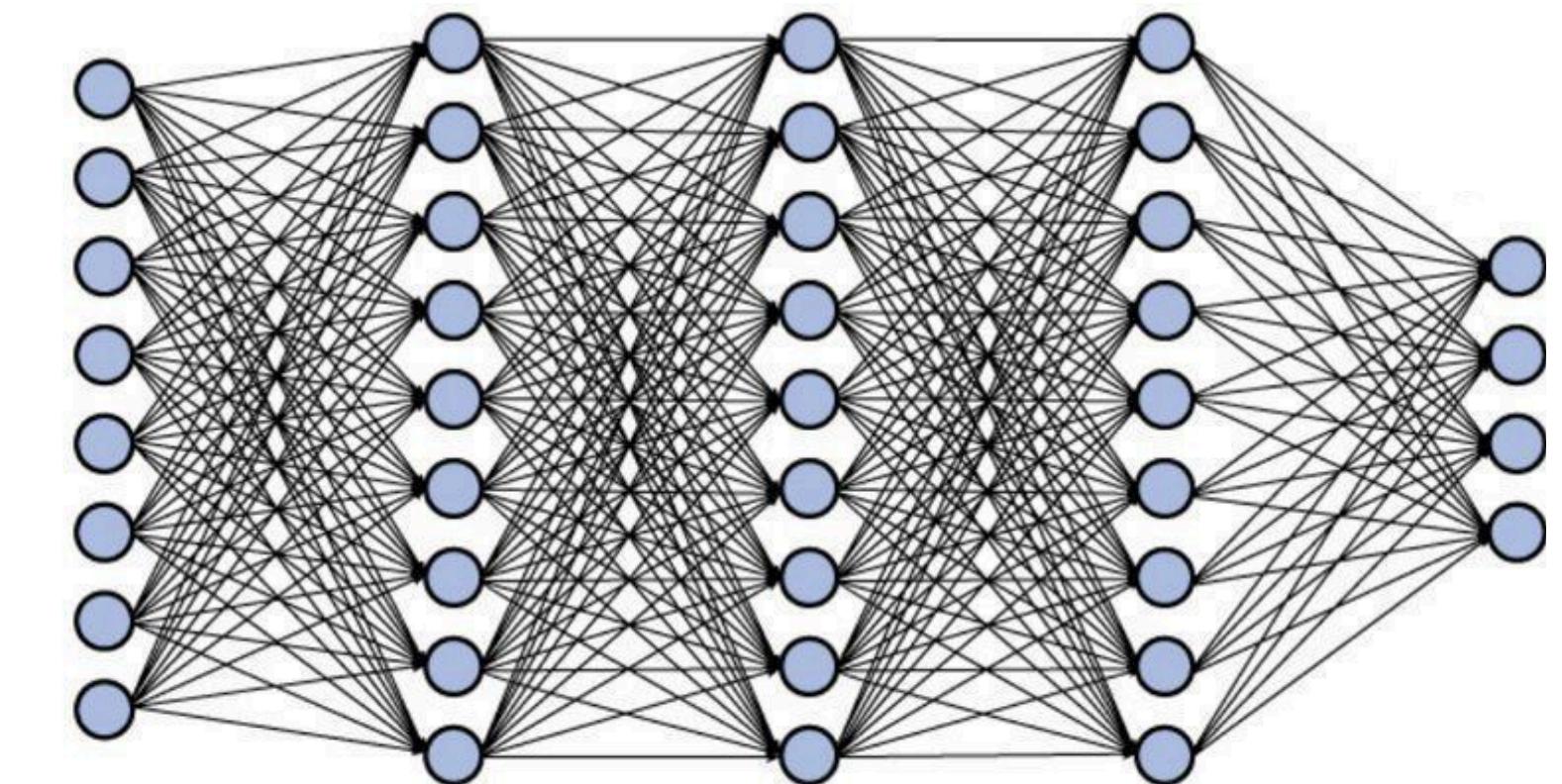
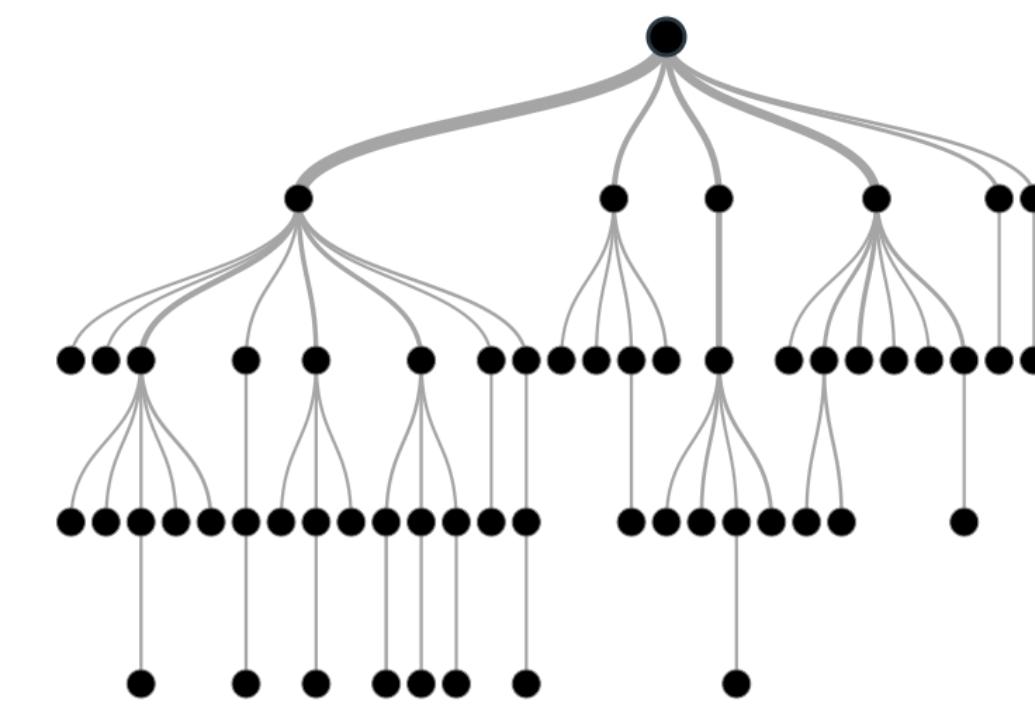
Learning type



Inherently interpretable

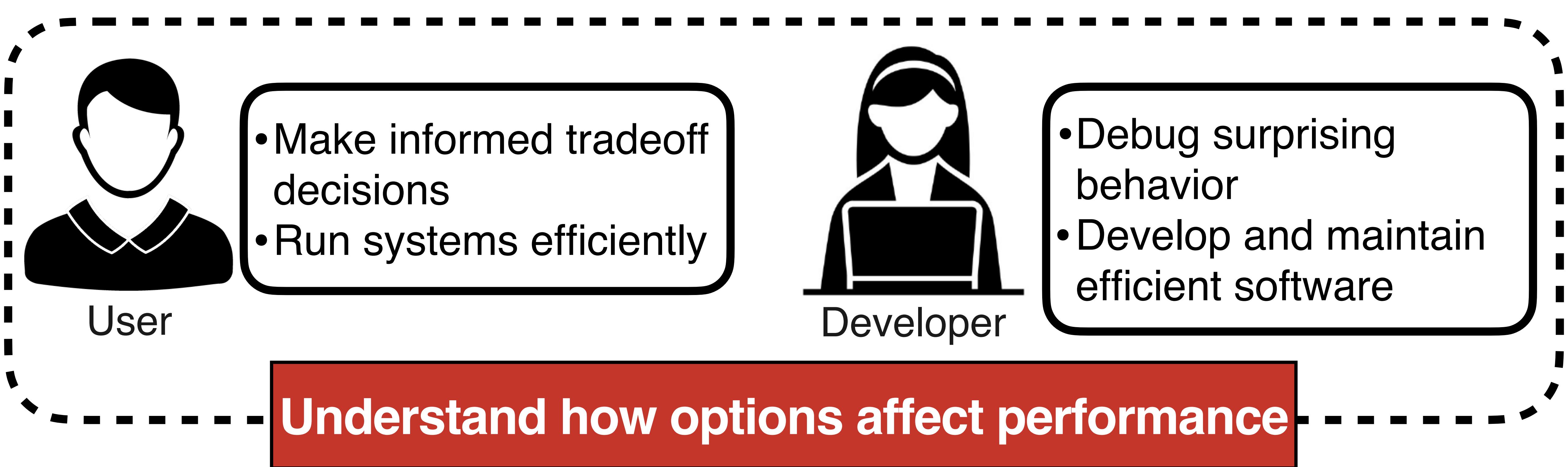
$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption

Not inherently interpretable



$$T = 25$$

$$\begin{aligned} &+ 3 \cdot \text{Logging} \\ &- 5 \cdot \text{Caching} \cdot \text{Interrupt} \\ &+ 9 \cdot \text{Sequential} \end{aligned}$$



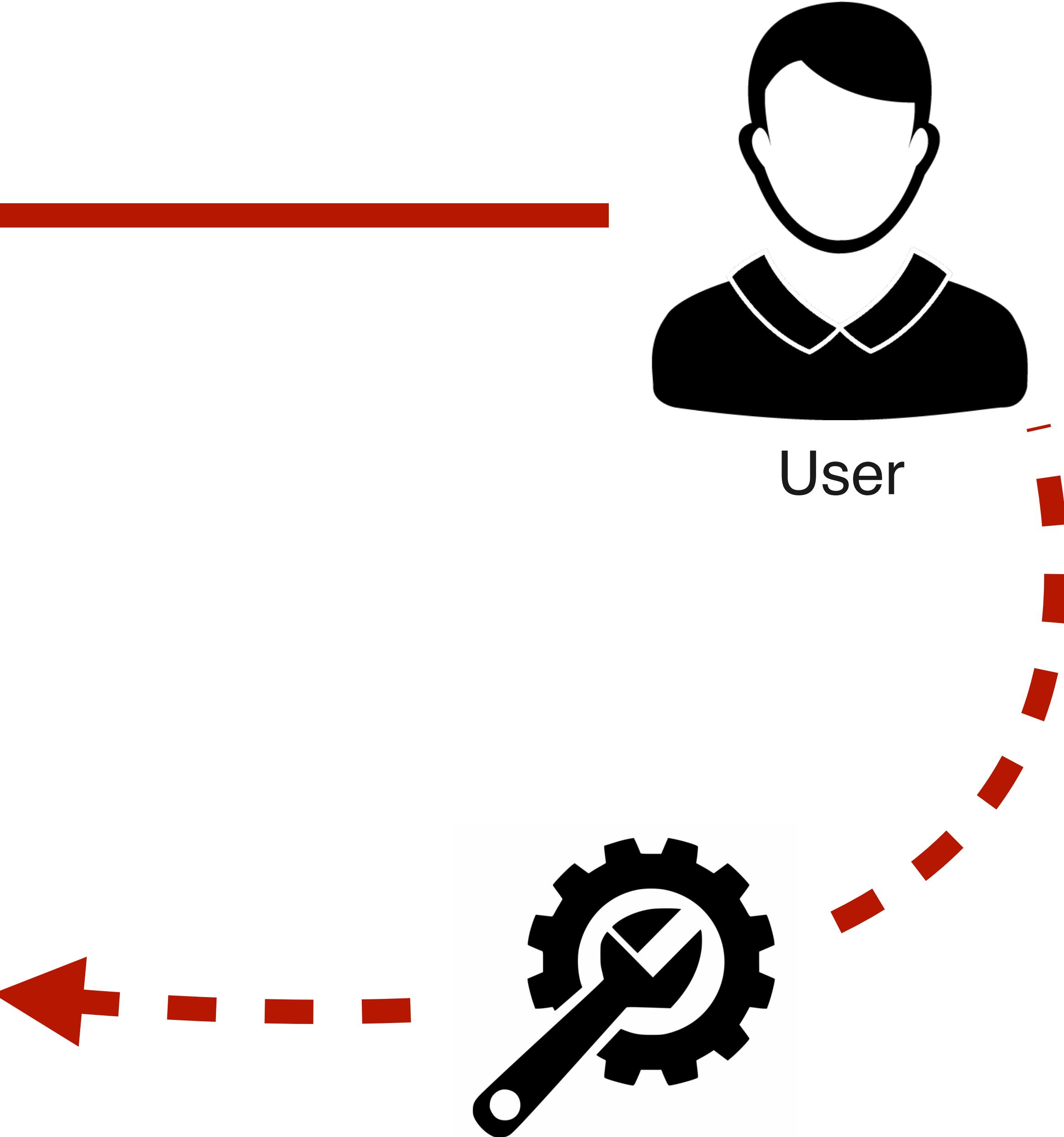
$T = 25$

- + 3 · Logging
- 5 · Indexing
- + 9 · Compression · Encryption



System

$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption



Make trade off decisions

+ 25% Logging

- 5% Indexing

+ 9% Compression · Encryption



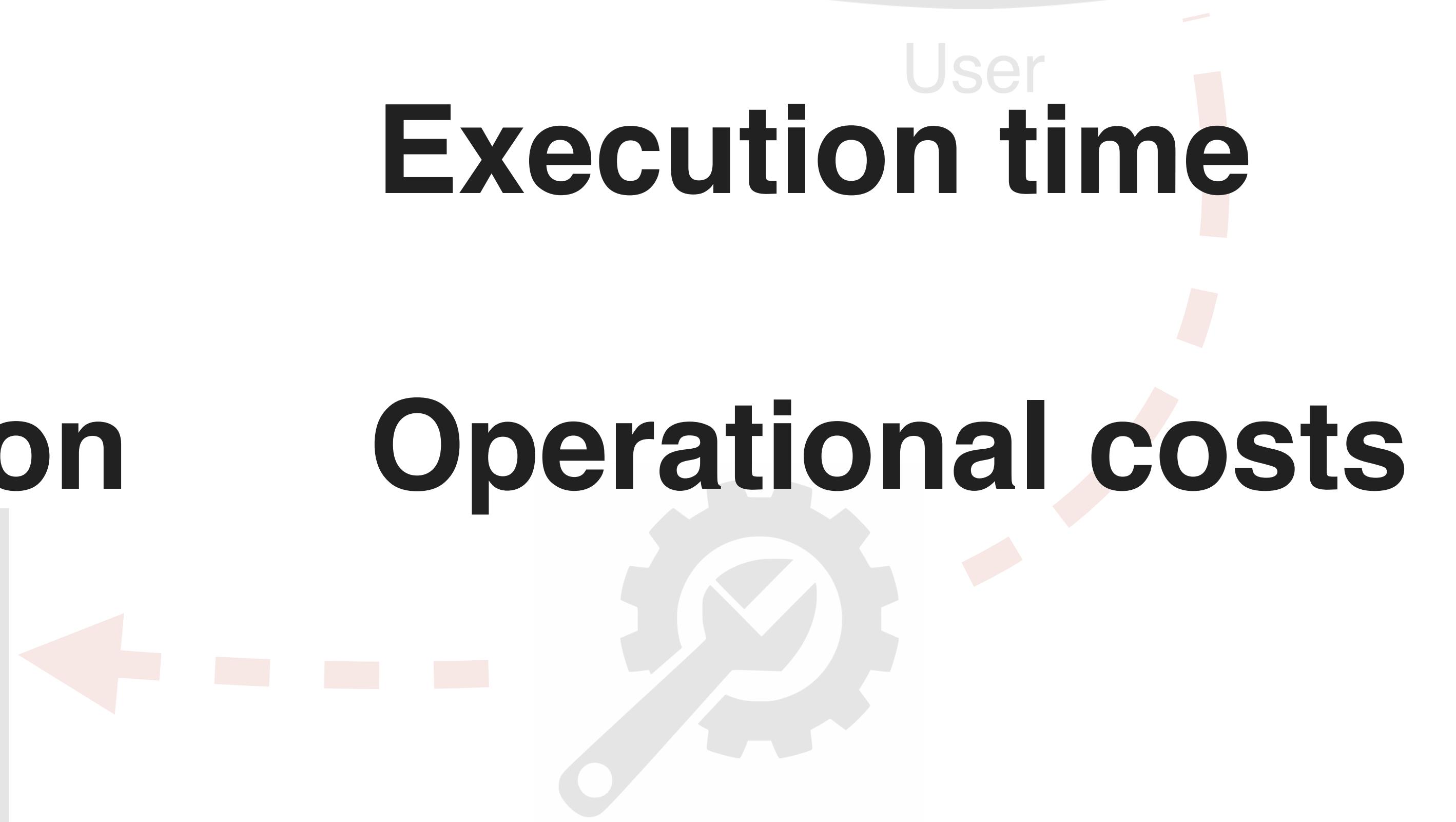
Performance

Energy consumption

System

Execution time

Operational costs



$T = 25$

- + 3 · Logging
- 5 · Indexing
- + 9 · Compression · Encryption



T = 25

+ 3 · Logging

- 5 · Indexing

• Encryption

Build



Developer

$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption



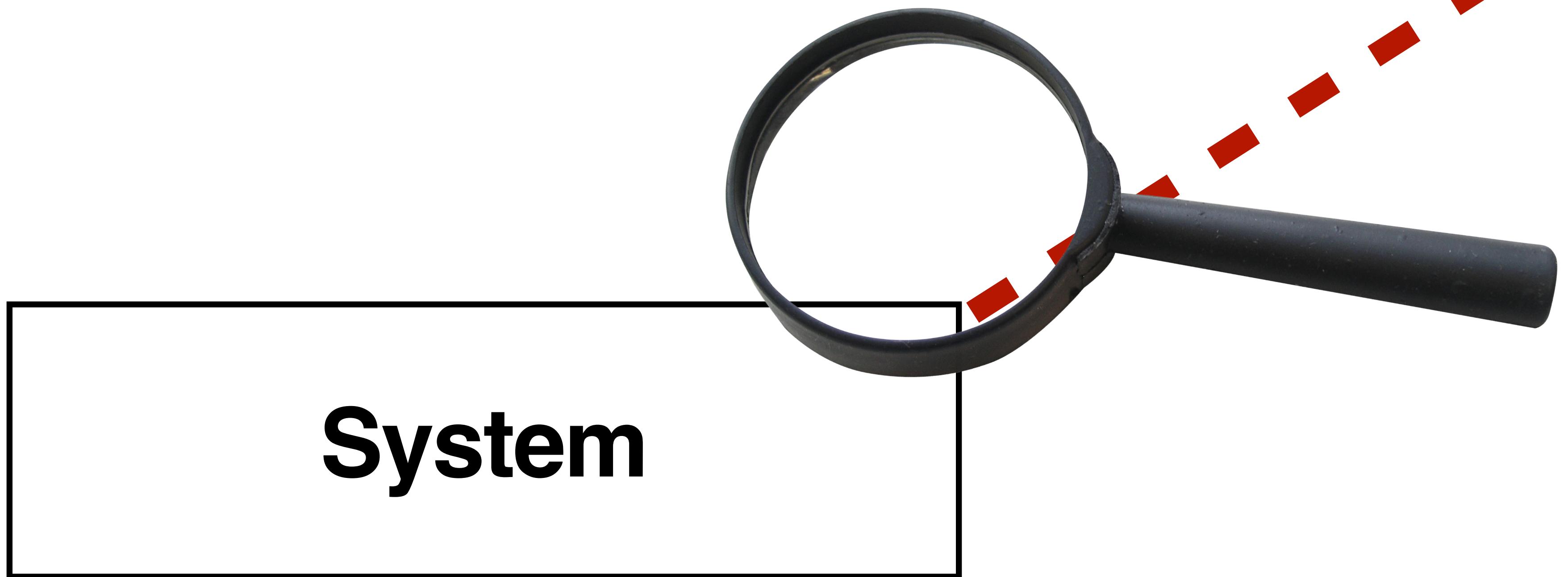
System

$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption

System

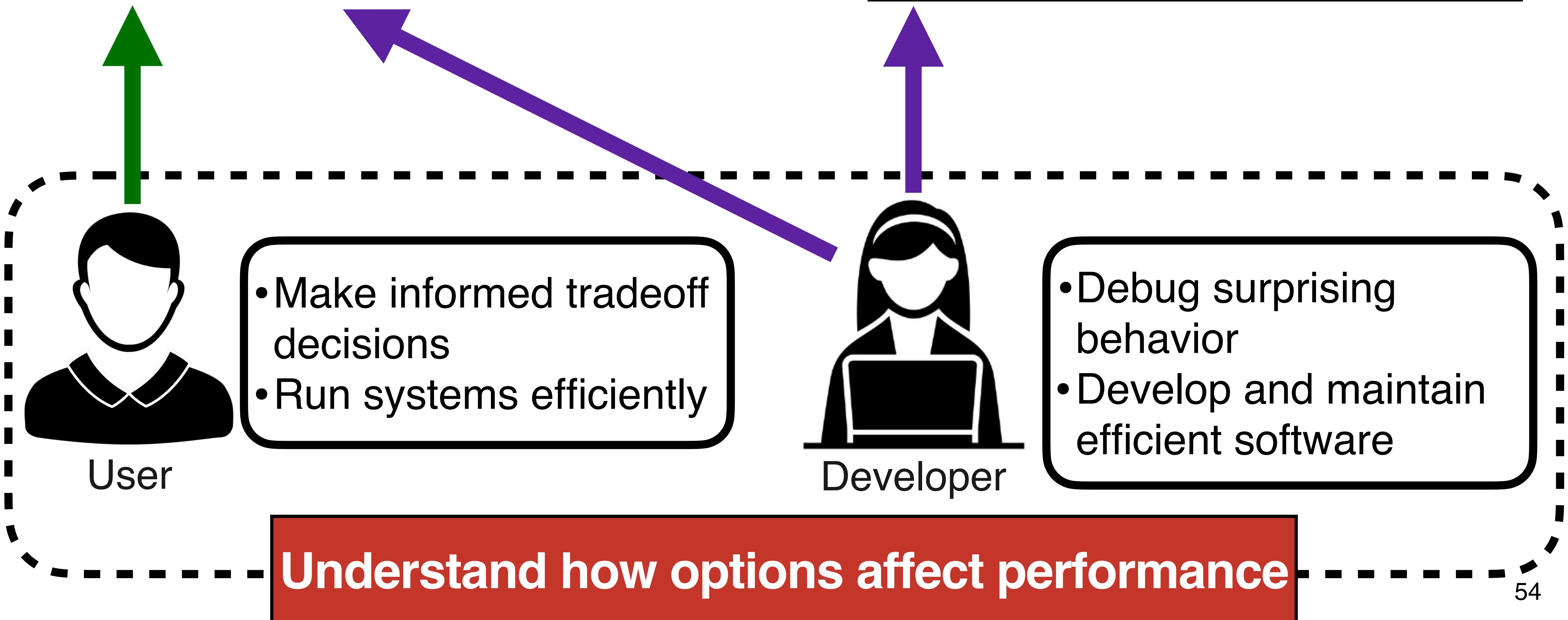


$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption



System

System



Thesis Statement

White-box analysis of how options influence the performance of code-level structures in configurable systems:

- (1) helps to **efficiently build accurate and interpretable global and local performance-influence models**
- (2) guides developers to **inspect, understand, and debug configuration-related performance behaviors.**

Thesis Statement

White-box analysis of how options influence the performance of code-level structures in configurable systems:

- (1) helps to **efficiently build accurate and interpretable global and local performance-influence models**
- (2) guides developers to **inspect, understand, and debug configuration-related performance behaviors.**

Thesis Statement

White-box analysis of how options influence the performance of code-level structures in configurable systems:

- (1) helps to **efficiently build accurate and interpretable global and local performance-influence models**
- (2) guides developers to **inspect, understand, and debug configuration-related performance behaviors.**

Thesis Statement

White-box analysis of how options influence the performance of code-level structures in configurable systems:

- (1) helps to **efficiently build accurate and interpretable global and local performance-influence models**
- (2) guides developers to **inspect, understand, and debug configuration-related performance behaviors.**

Thesis Statement

White-box analysis of how options influence the performance of code-level structures on configurable systems:

(1) helps to efficiently build accurate and interpretable global and local performance-influence models

(2) guides developers to understand, and debug configuration-related performance behavior.

But why

white-box?



Insights!

Not all options tend to affect the performance for all workloads

Few options tend to interact in configurable systems



Insight!

Performance in configurable systems tends to change at control-flow statements

WHY PROGRAMS FAIL

A GUIDE TO SYSTEMATIC DEBUGGING

The screenshot shows a Java application running in a debugger. The code editor displays a class named `ImageHandler` with several methods and annotations. A tooltip is visible over the variable `compressionList`. The debugger's frames view shows the current thread is "pool-1... RUNNING". The variables pane shows the `args` object with properties like `src`, `dst`, `scale`, `platform`, `outputCompressionMode`, `scaleMode`, and `downScalingAlgorithm`. The memory analysis pane indicates no classes are loaded.

Debugging requires analyzing the implementation



User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

1

Modeling

Global

Local



User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

1

Modeling

Global

Local

2

Debugging



User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

1

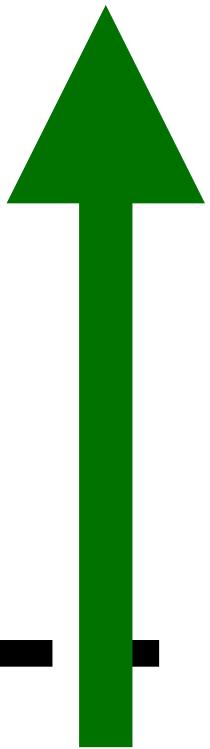
Modeling

Global

Local

2

Debugging



User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

1

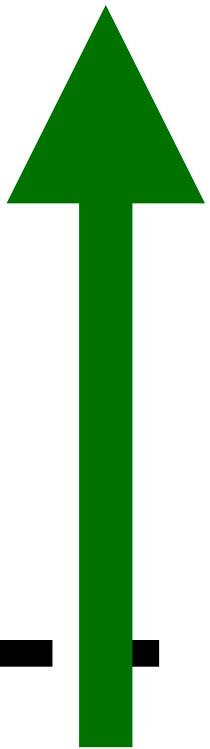
Modeling

Global

Local

2

Debugging



User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

1

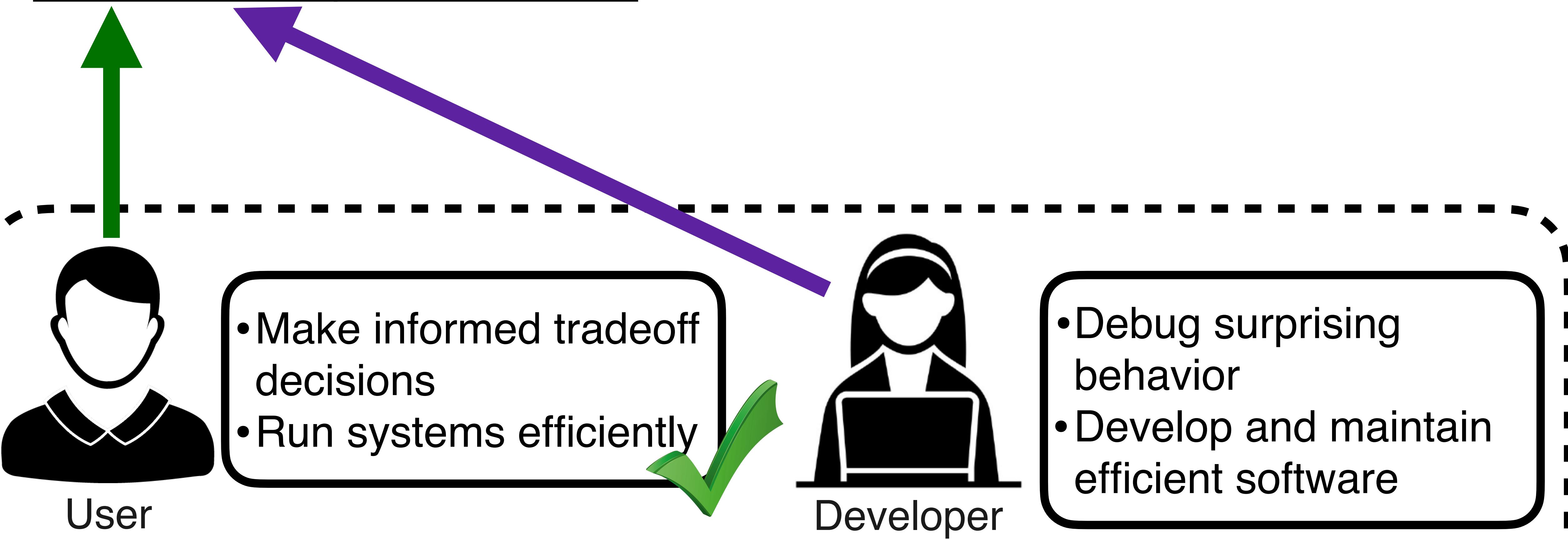
Modeling

Global

Local

2

Debugging



Understand how options affect performance

1

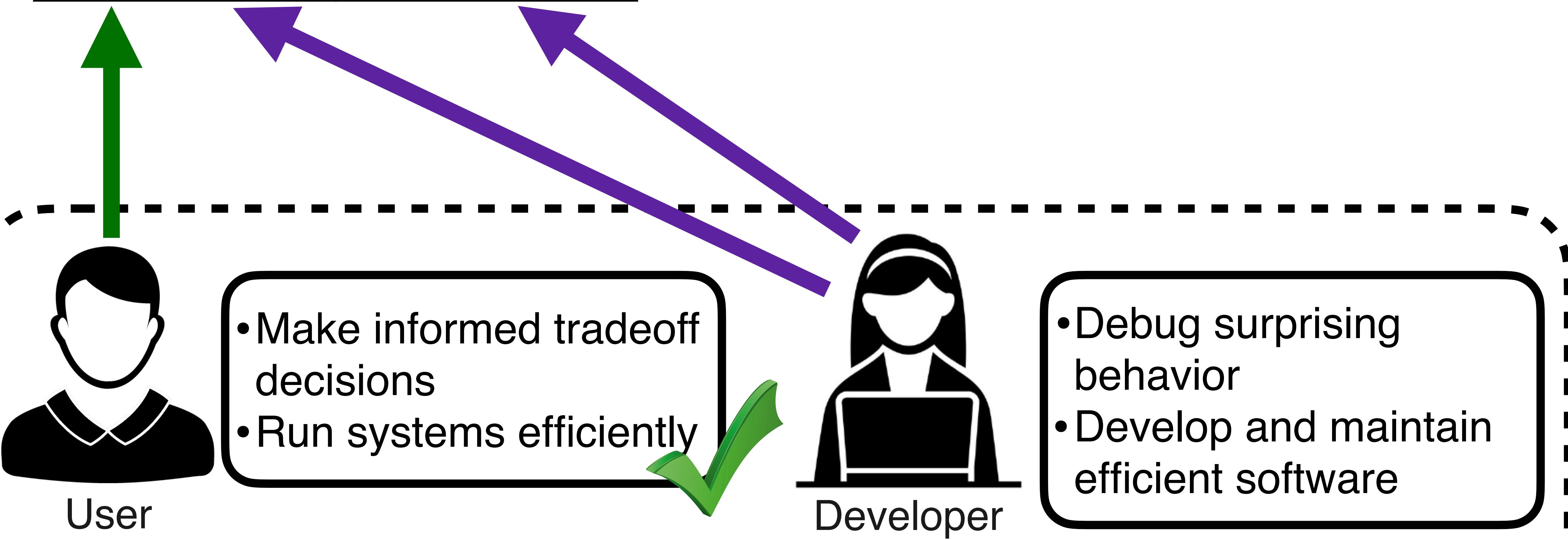
Modeling

Global

Local

2

Debugging



Understand how options affect performance

1

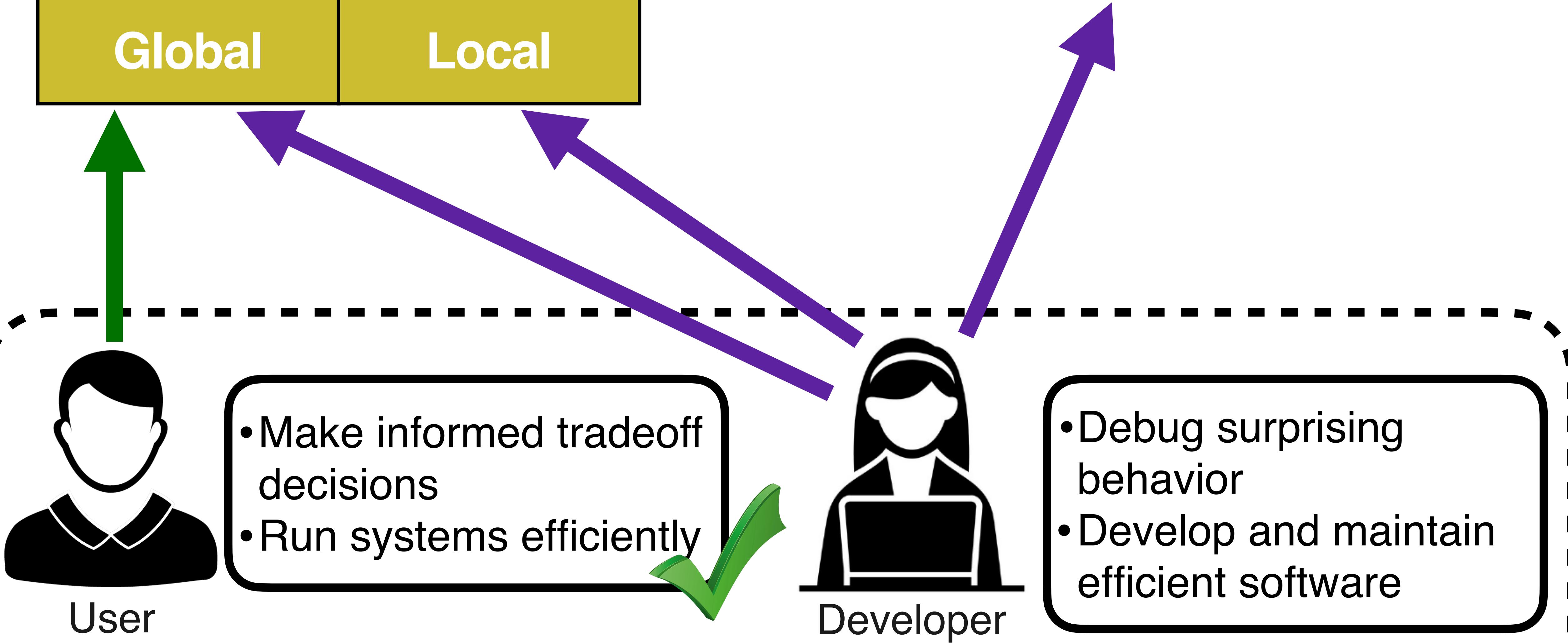
Modeling

Global

Local

2

Debugging



Understand how options affect performance

1

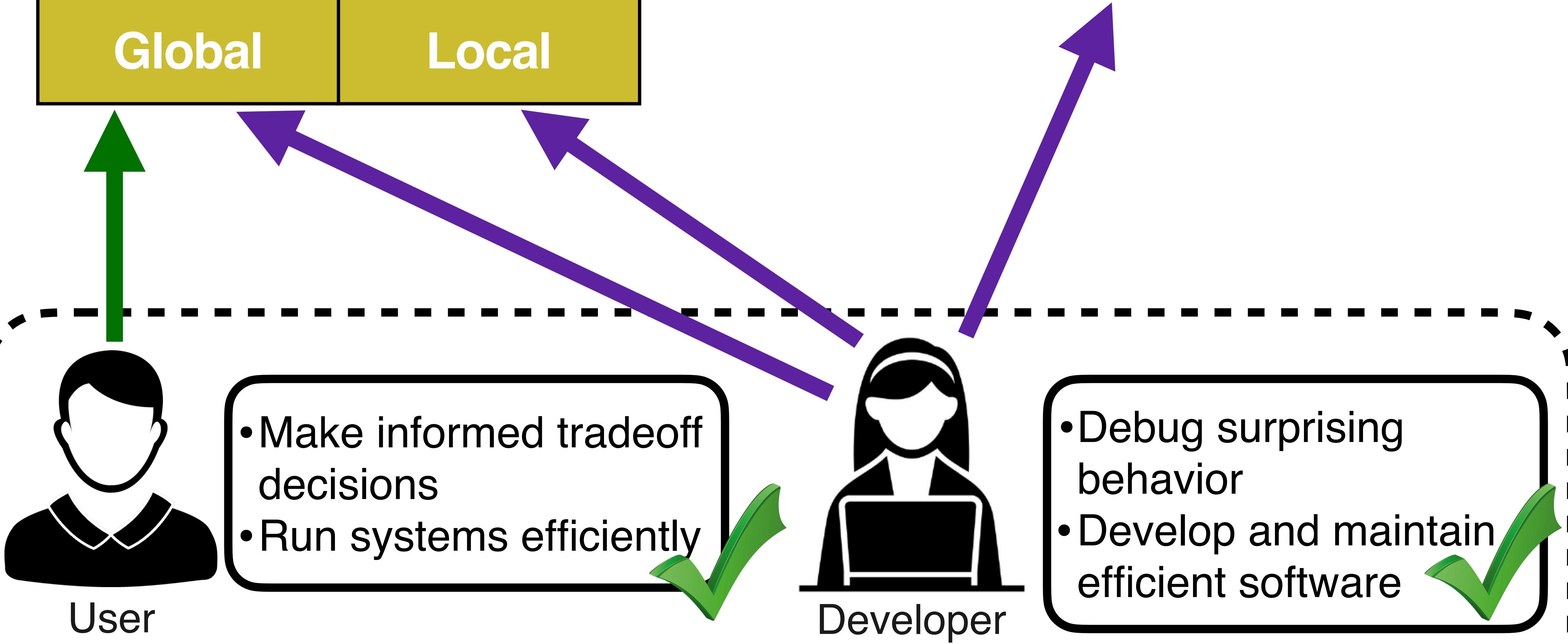
Modeling

Global

Local

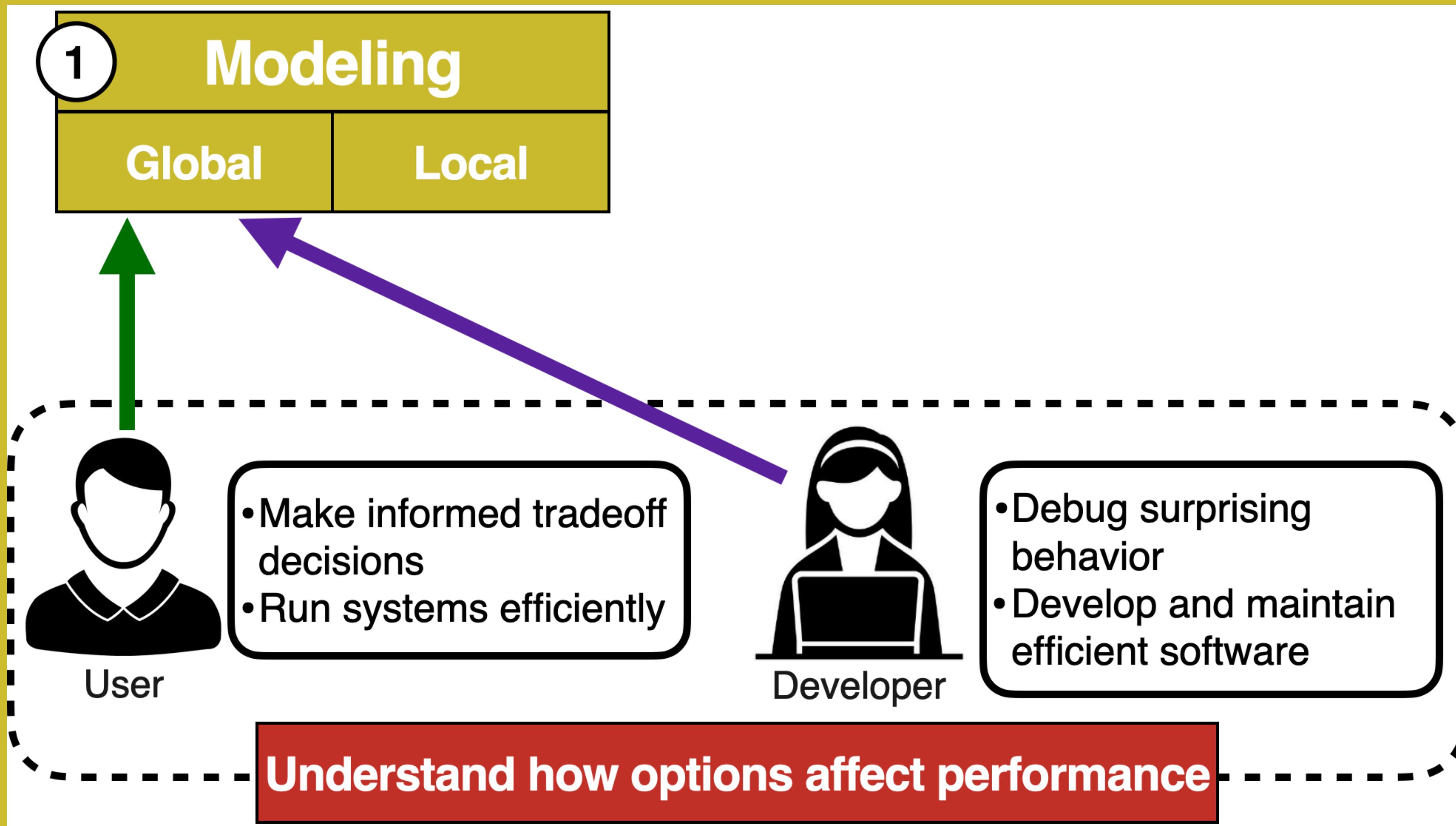
2

Debugging



Understand how options affect performance

White-box Performance Modeling of Configurable Systems



```
1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x)
10         convert(y);
11     ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x)
15         ... // execution time: 3 seconds
16     else
17         ... // execution time: 2 seconds
18 }
```

```
1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x)
10         convert(y);
11     ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x)
15         ... // execution time: 3 seconds
16     else
17         ... // execution time: 2 seconds
18 }
```



Load options

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interp");  
4     boolean c = getopt("Sequential");  
5     ...  
6     process(a, b);  
7 }  
8 def process(boolean x, boolean y) {  
9     if (x)  
10        convert(y);  
11    ... // execution time: 5 seconds  
12 }  
13 def convert(boolean x) {  
14     if (x)  
15        ... // execution time: 3 seconds  
16     else  
17        ... // execution time: 2 seconds  
18 }
```

Propagate options

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if (x)
10        convert(y),
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if (x)
15        ... // execution time: 3 seconds
16     else
17        ... // execution time: 2 seconds
18 }

```

Use options

```
1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x)
10         convert(y);
11     ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x)
15         ... // execution time: 3 seconds
16     else
17         ... // execution time: 2 seconds
18 }
```



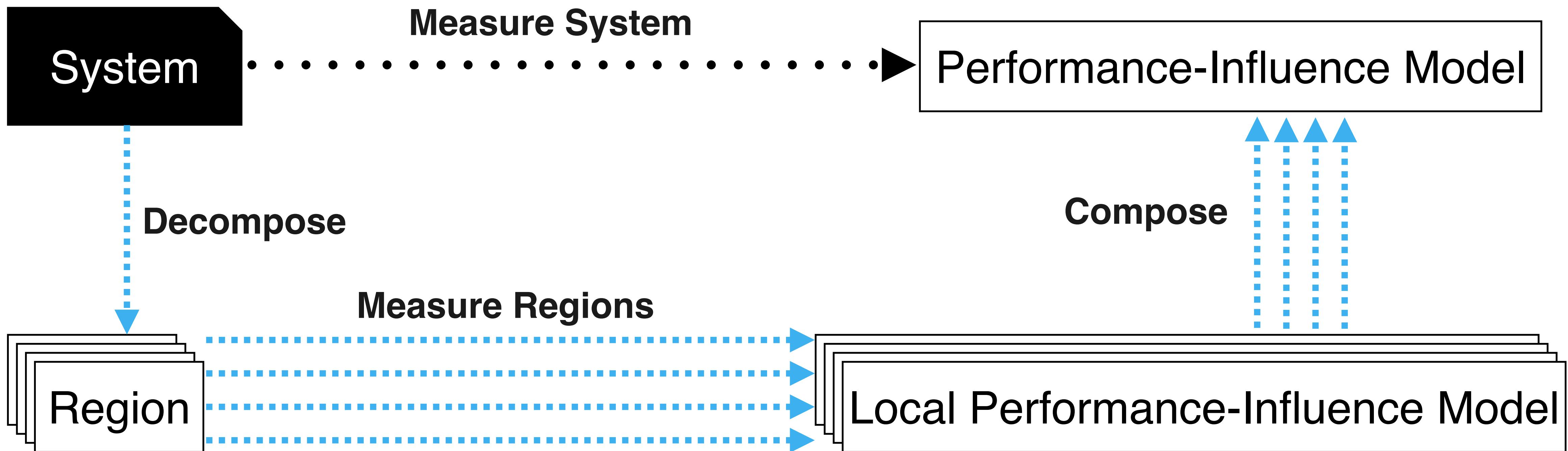
Our insights!

Compositionality

Compression

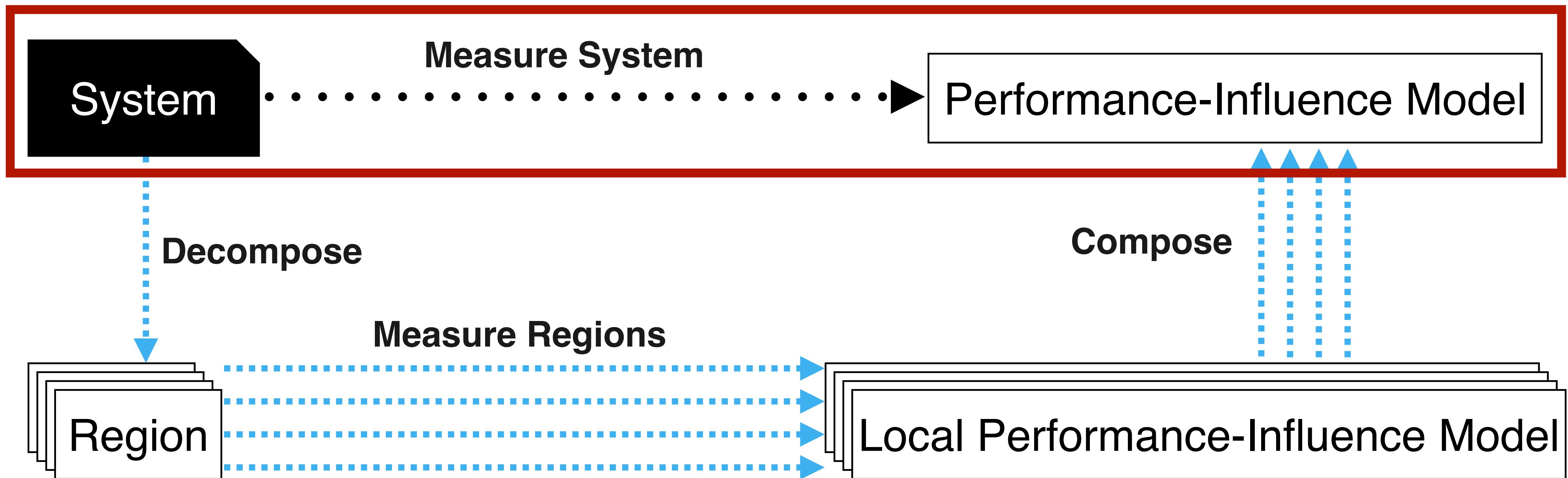
Compositionality

Performance-influence models can be built by composing models built independently for smaller regions of code



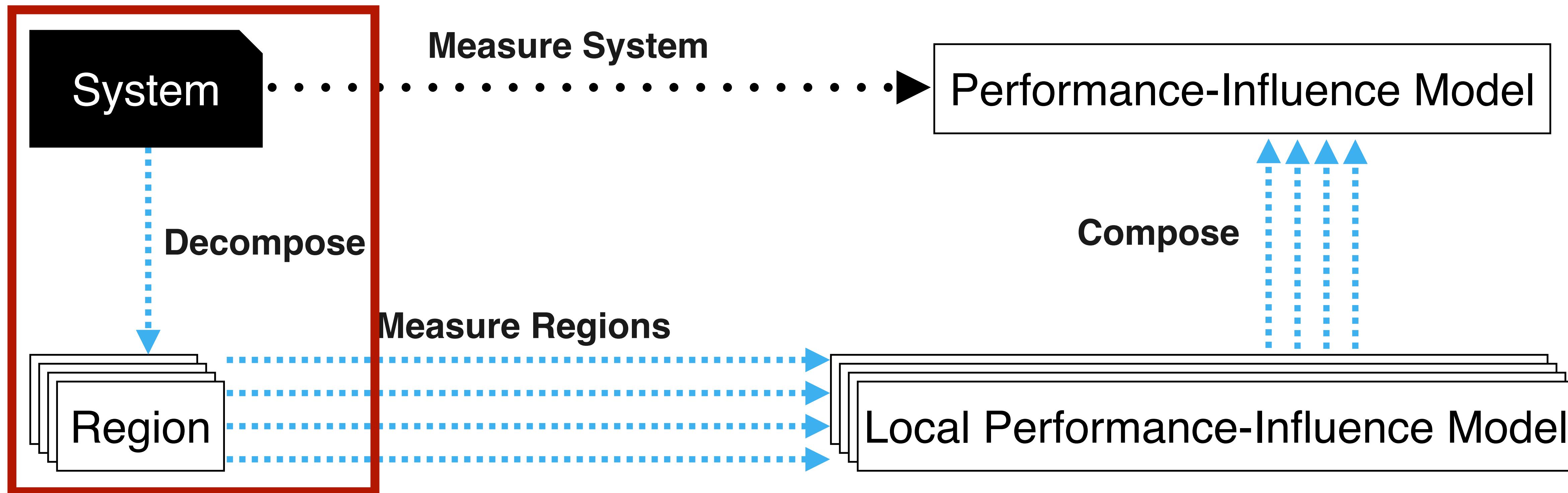
Compositionality

Performance-influence models can be built by composing models built independently for smaller regions of code



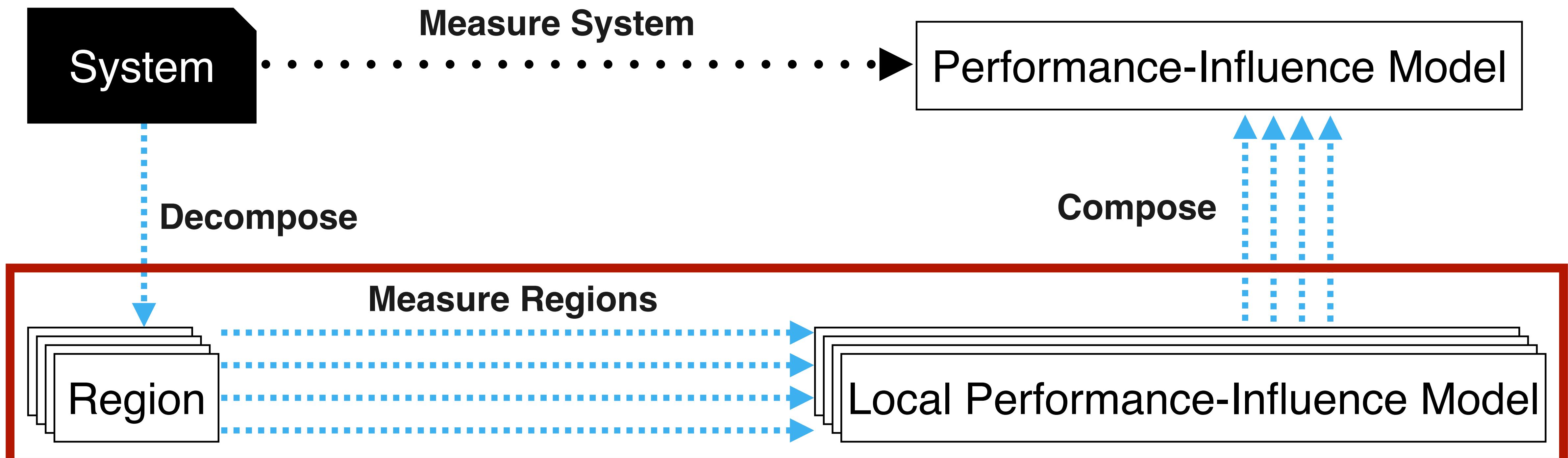
Compositionality

Performance-influence models can be built by composing models built independently for smaller regions of code



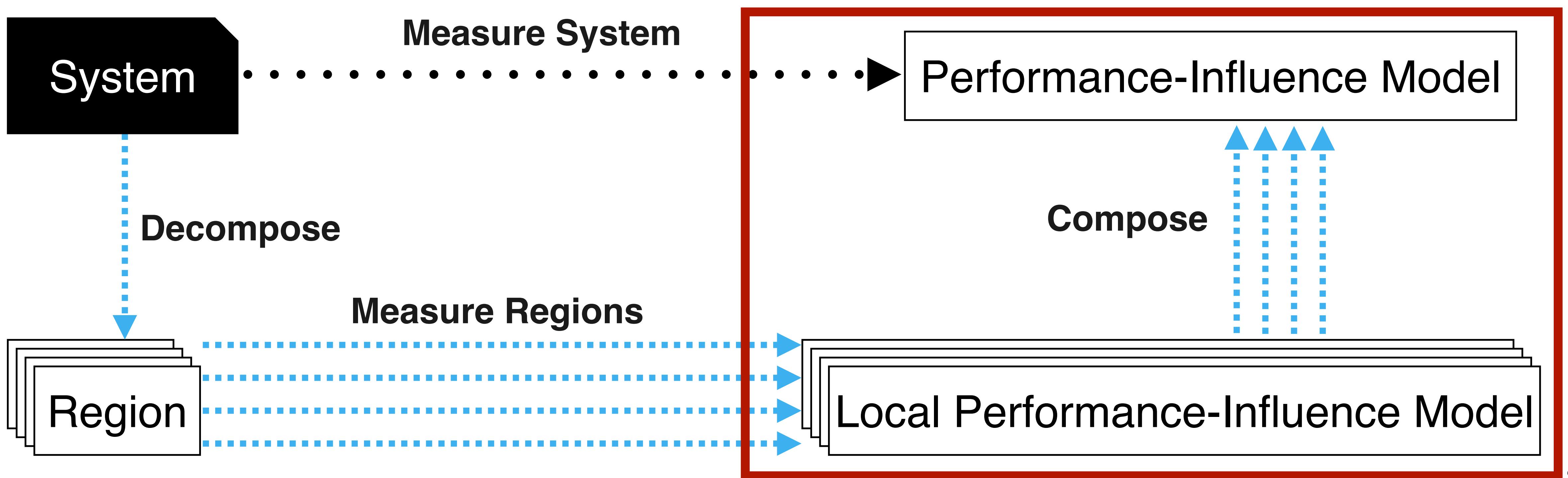
Compositionality

Performance-influence models can be built by composing models built independently for smaller regions of code



Compositionality

Performance-influence models can be built by composing models built independently for smaller regions of code



```
1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x)
10         convert(y);
11     ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x)
15         ... // execution time: 3 seconds
16     else
17         ... // execution time: 2 seconds
18 }
```

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     process(a, b);  
7 }  
8 def process(boolean x, boolean y) {  
9     if(x)  
10         convert(y);  
11     ... // execution time: 5 seconds  
12 }  
13 def convert(boolean x) {  
14     if(x)  
15         ... // execution time: 3 seconds  
16     else  
17         ... // execution time: 2 seconds  
18 }
```

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16 else
17    ... // execution time: 2 seconds
18 }

```

Caching	Time
TRUE	
FALSE	

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16 else
17    ... // execution time: 2 seconds
18 }

```

Caching	Time
TRUE	5
FALSE	

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16 else
17    ... // execution time: 2 seconds
18 }

```

Caching	Time
TRUE	5
FALSE	0

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16 else
17    ... // execution time: 2 seconds
18 }

```

Caching	Time
TRUE	5
FALSE	0

$$T_{process} = 5 \cdot \text{Caching}$$

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16     else
17        ... // execution time: 2 seconds
18 }

```

$$T_{\text{process}} = 5 \cdot \text{Caching}$$

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16     else
17        ... // execution time: 2 seconds
18 }

```

$$T_{\text{process}} = 5 \cdot \text{Caching}$$

Caching	Interrupt	Time
TRUE	TRUE	3
TRUE	FALSE	2

$$T_{\text{convert}} = 2 \cdot \text{Caching} + 1 \cdot \text{Caching} \cdot \text{Interrupt}$$

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16     else
17        ... // execution time: 2 seconds
18 }

```

$$T_{\text{process}} = 5 \cdot \text{Caching}$$

$$T_{\text{convert}} = 2 \cdot \text{Caching} \\ + 1 \cdot \text{Caching} \cdot \text{Interrupt}$$

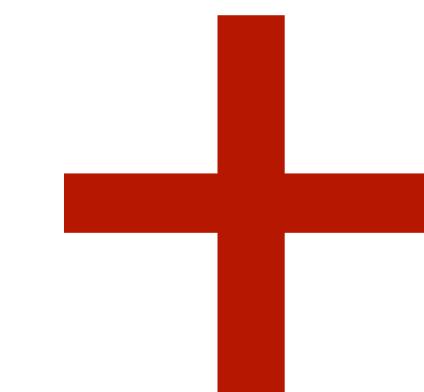
```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16     else
17        ... // execution time: 2 seconds
18 }

```

Compositionality

$$T_{\text{process}} = 5 \cdot \text{Caching}$$



$$T_{\text{convert}} = 2 \cdot \text{Caching} + 1 \cdot \text{Caching} \cdot \text{Interrupt}$$

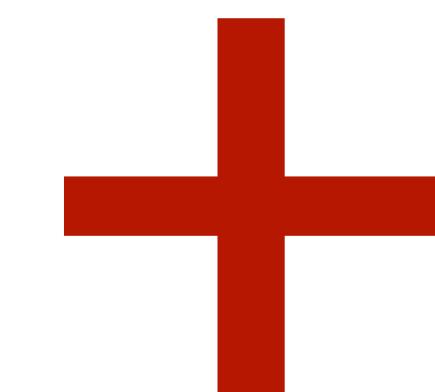
```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10        convert(y);
11        ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15        ... // execution time: 3 seconds
16     else
17        ... // execution time: 2 seconds
18 }

```

Compositionality

$$T_{\text{process}} = 5 \cdot \text{Caching}$$



$$T_{\text{convert}} = 2 \cdot \text{Caching} + 1 \cdot \text{Caching} \cdot \text{Interrupt}$$

$$T = 7 \cdot \text{Caching} + 1 \cdot \text{Caching} \cdot \text{Interrupt}$$

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     process(a, b);
6 }
7 }

8 def process(boolean x, boolean y) {
9     if(x) // region depends on Caching
10    convert(y);
11    ... // execution time: 5 seconds
12 }

13 def convert(boolean x) {
14     if(x) // region depends on Interrupt
15     ...
16     else // execution time: 3 seconds
17     ...
18 }

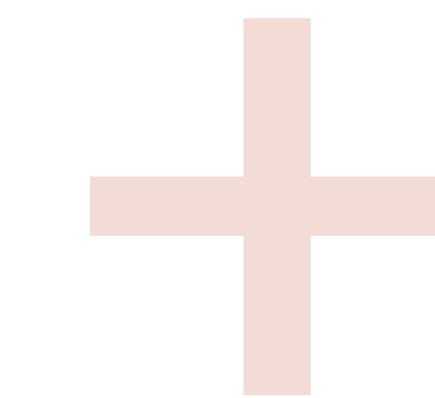
```

Insight!

Compositionality



$$T_{process} = 5 \cdot \text{Caching}$$



$$T_{convert} = 3 \cdot \text{Caching} + 1 \cdot \text{Caching} \cdot \text{Interrupt}$$

Few options tend to interact in smaller regions

$$T = 7 \cdot \text{Caching} + 1 \cdot \text{Caching} \cdot \text{Interrupt}$$



Our insights!

Compositionality

Compression

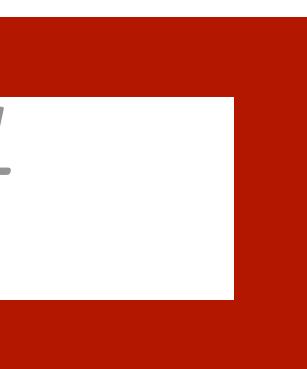
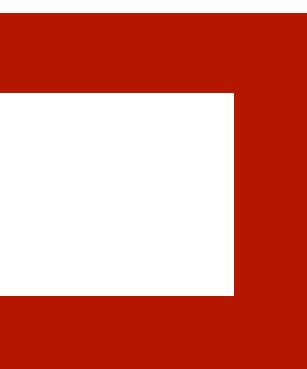
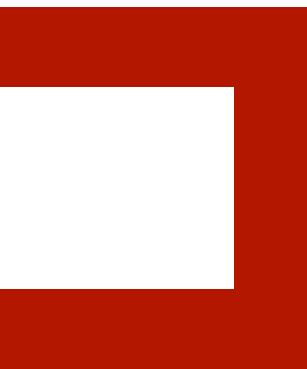
Compression

Compression allow us to simultaneously explore paths in multiple independent regions with a few configurations

Compression

Caching Interrupt Sequential

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     if(a) // variable depends on Caching  
7         ... // execution time: 1 second  
8     if(b) // variable depends on Interrupt  
9         ... // execution time: 2 seconds  
10    if(c) // variable depends on Sequential  
11        ... // execution time: 3 seconds  
12 }
```



Caching	Time
TRUE	
FALSE	

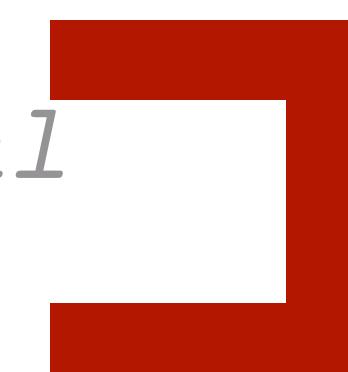
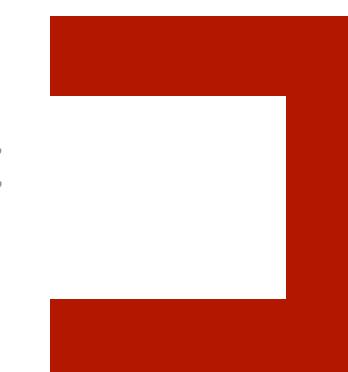
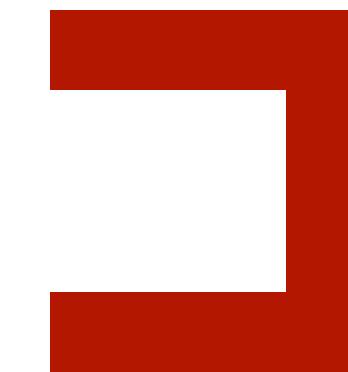
Interrupt	Time
TRUE	
FALSE	

Sequential	Time
TRUE	
FALSE	

Compression

Caching	Interrupt	Sequential
FALSE	FALSE	FALSE

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     if(a) // variable depends on Caching  
7         ... // execution time: 1 second  
8     if(b) // variable depends on Interrupt  
9         ... // execution time: 2 seconds  
10    if(c) // variable depends on Sequential  
11        ... // execution time: 3 seconds  
12 }
```



Caching	Time
TRUE	
FALSE	0

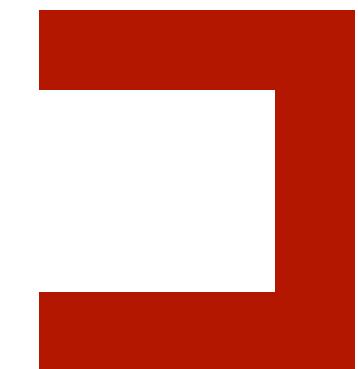
Interrupt	Time
TRUE	
FALSE	0

Sequential	Time
TRUE	
FALSE	0

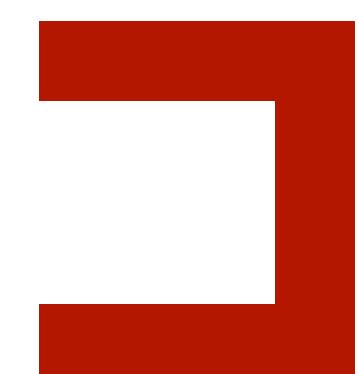
Compression

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     if(a) // variable depends on Caching  
7         ... // execution time: 1 second  
8     if(b) // variable depends on Interrupt  
9         ... // execution time: 2 seconds  
10    if(c) // variable depends on Sequential  
11        ... // execution time: 3 seconds  
12 }
```

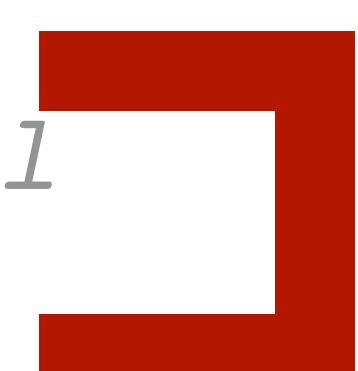
Caching	Interrupt	Sequential
FALSE	FALSE	FALSE
TRUE	TRUE	TRUE



Caching	Time
TRUE	1
FALSE	0

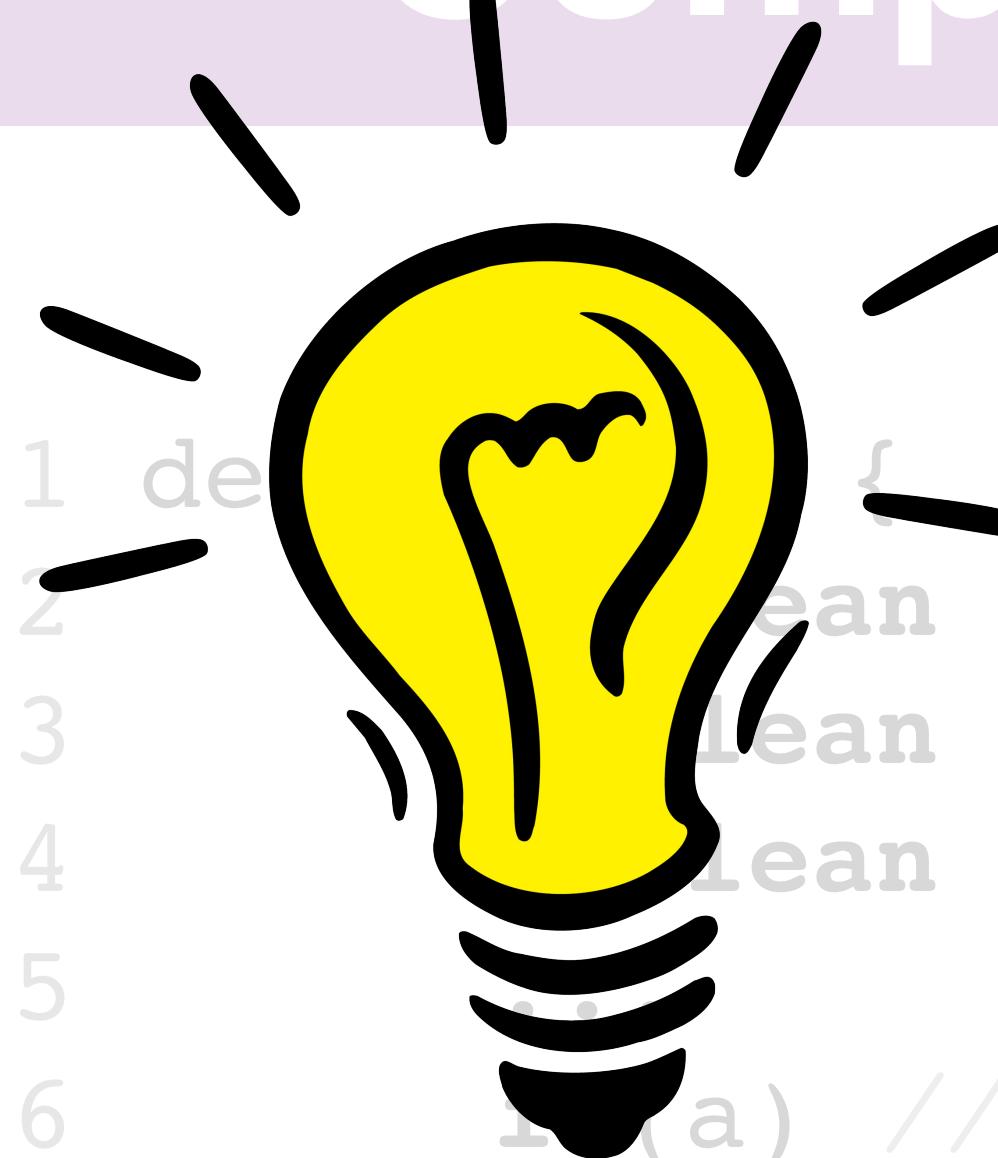


Interrupt	Time
TRUE	2
FALSE	0



Sequential	Time
TRUE	3
FALSE	0

Compression



Insight!

```
1  de
2  lean a = getopt("Caching");
3  lean b = getopt("Interrupt");
4  lean c = getopt("Sequential");
5
6  if(a) // variable depends on Caching
7  ...
8  if(b) // variable depends on Interrupt
9  ...
10 if(c) // variable depends on Sequential
11 ...
12 }
```



Caching	Interrupt	Sequential
FALSE	FALSE	FALSE
TRUE	TRUE	TRUE

Caching	Time
TRUE	1
FALSE	0

Interrupt	Time
TRUE	2
FALSE	0

Sequential	Time
TRUE	3
FALSE	0

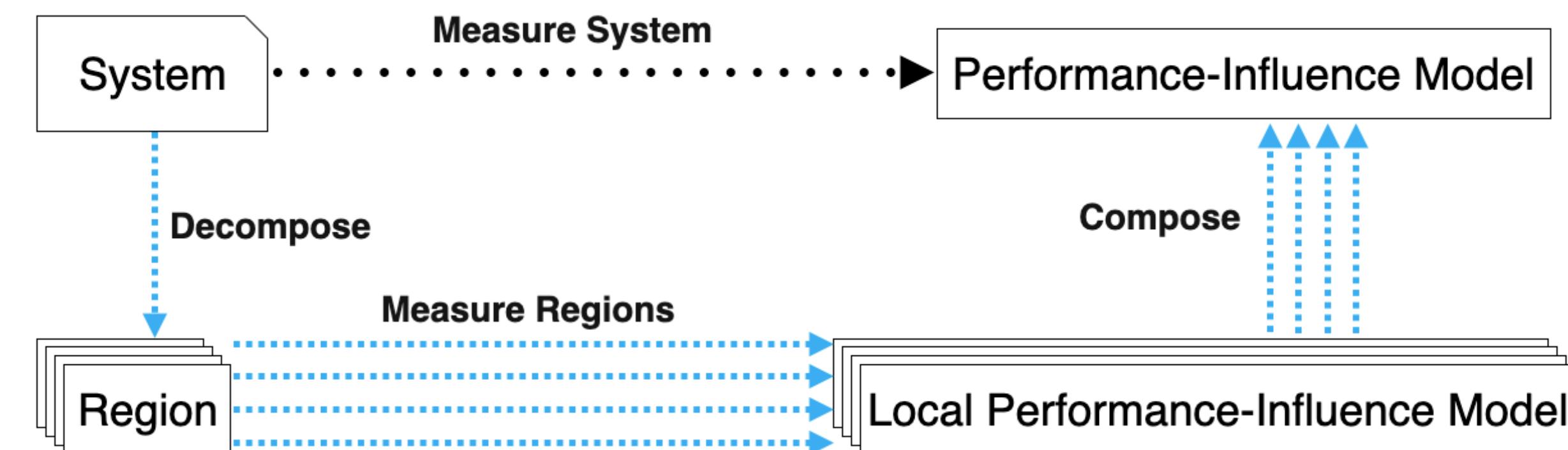
Few different options tend to interact in different regions

Compositionality

Compression



Our insights!



```
1 def main() {  
2     boolean a = getOpt("Caching");  
3     boolean b = getOpt("Interrupt");  
4     boolean c = getOpt("Sequential");  
5     ...  
6     if(a) // variable depends on Caching  
        ... // execution time: 1 second  
7  
8     if(b) // variable depends on Interrupt  
        ... // execution time: 2 seconds  
9  
10    if(c) // variable depends on Sequential  
        ... // execution time: 3 seconds  
11  
12 }
```

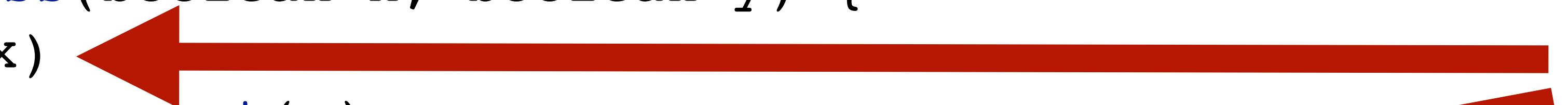
Caching	Interrupt	Sequential
FALSE	FALSE	FALSE
TRUE	TRUE	TRUE

Caching	Time
TRUE	1
FALSE	0

Interrupt	Time
TRUE	2
FALSE	0

Sequential	Time
TRUE	3
FALSE	0

Taint Analysis

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     process(a, b);  
7 }  
8 def process(boolean x, boolean y) {  
9     if (x)  convert(y);  
10    ... // execution time: 5 seconds  
11 }  
12 def convert(boolean x, {  
13     if (x)  ... // execution time: 3 seconds  
14     else  ... // execution time: 2 seconds  
15 }  
16 else  
17 ... // execution time: 2 seconds  
18 }
```

```
1 def main() {  
2     boolean a = getopt('Caching');  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     process(a, b);  
7 }  
8 def process(boolean x, boolean y) {  
9     if (x)  
10        convert(y);  
11    ... // execution time: 5 seconds  
12 }  
13 def convert(boolean x) {  
14     if (x)  
15        ... // execution time: 3 seconds  
16     else  
17        ... // execution time: 2 seconds  
18 }
```

Caching

Caching

Caching

Propagate taints

The diagram illustrates the flow of data and taint propagation in the given Java code. Blue curved arrows labeled "Caching" show the flow of values from the `getOpt` calls through the `process` method to the `convert` method. A red arrow labeled "Propagate taints" shows the propagation of taint from the `convert` method back to the `process` method.

```
1 def main() {  
2     boolean a = getopt('Caching');  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     process(a, b);  
7 }  
8 def process(boolean x, boolean y) {  
9     if (x) convert(y);  
10    ... // execution time: 5 seconds  
11 }  
12 }  
13 def convert(boolean x) {  
14     if (x)  
15        ... // execution time: 3 seconds  
16    else  
17        ... // execution time: 2 seconds  
18 }
```

Caching

Caching

Caching

Caching

Caching

Caching

Influencing options

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if (x)
10        convert(y);
11     ... // execution time: 5 seconds
12 }
13 def convert(boolean x) {
14     if (x)
15         ... // execution time: 3 seconds
16     else
17         ... // execution time: 2 seconds
18 }

```

Caching	Time
TRUE	
FALSE	



Configurations to measure

```

1 def main() {
2     boolean a = getopt("Caching");
3     boolean b = getopt("Interrupt");
4     boolean c = getopt("Sequential");
5     ...
6     process(a, b);
7 }
8 def process(boolean x, boolean y) {
9     if (x)
10        convert(y);
11    ...
12    // execution time: 5 seconds
13 }
14 def convert(boolean x) {
15     if (x)
16        ...
17        // execution time: 3 seconds
18 }

```

// execution time: 2 seconds

Tracking implicit flows

Analyze Regions

Taint Analysis

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     process(a, b);  
7 }  
8 def process(boolean x, boolean y) {  
9     if(x) convert(y);  
10    ... // execution time: 5 seconds  
11 }  
12 def convert(boolean x) {  
13     if(x) ...  
14     else ... // execution time: 3 seconds  
15 }  
16 ... // execution time: 2 seconds  
17 }  
18 }
```

Annotations: Caching arrows point from `getOpt` to `process`, `process` to `convert`, and `convert` to its body. A red arrow points from `convert` back to `process`. A note indicates `... // execution time: 5 seconds`.

Measure Performance

Compression

Caching	Interrupt	Sequential
FALSE	FALSE	FALSE
TRUE	TRUE	TRUE

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean b = getopt("Interrupt");  
4     boolean c = getopt("Sequential");  
5     ...  
6     if(a) // variable depends on Caching  
        ... // execution time: 1 second  
7     ...  
8     if(b) // variable depends on Interrupt  
        ... // execution time: 2 seconds  
9     ...  
10    if(c) // variable depends on Sequential  
        ... // execution time: 3 seconds  
11 }  
12 }
```

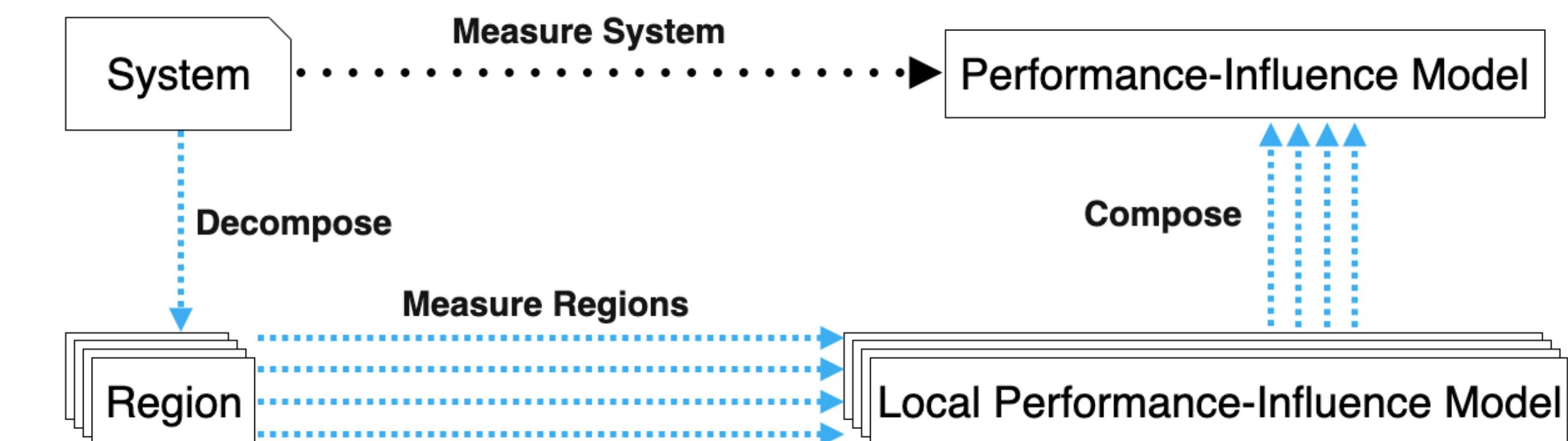
Caching	Time
TRUE	1
FALSE	0

Interrupt	Time
TRUE	2
FALSE	0

Sequential	Time
TRUE	3
FALSE	0

Build Model

Compositionality



ConfigCrusher

ASEJ'20

Static Taint Analysis

Regions: Control-flow
statements

Comprex

Under submission

(Iterative) Dynamic Taint Analysis

Regions: Methods

Both prototypes efficiently build accurate
and interpretable models

ASEJ'20

**Dynamic taint analysis scales to larger
systems**

Regions: Control-flow
statements

Under submission

(Iterative) Dynamic Taint Analysis

Regions: Methods

**Method-level granularity does not
sacrifice compression potential**

Evaluation

ConfigCrusher

ASEJ'20

Comprex

Under submission

32 combinations of sampling and machine learning approaches

13 open-source Java systems

Evaluate cost, accuracy, interpretability

Subject systems

System	# SLOC	# Options	# Configurations
Pngtastic Counter	1250	5	32
Pngtastic Optimizer	2553	5	32
Elevator	575	6	20
Grep	2152	7	128
Kanzi	20K	7	128
Email	696	9	40
Prevayler	1328	9	512
Sort	2163	12	4096
H2	142K	16	65K
Berkeley DB	164K	16	65K
Apache Lucene	396K	17	131K
Density Converter	49K	22	4.9M

Density Converter

22 options, 4.9 million configurations

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Complex
--------	--	-------------------------------	---------

Density Converter

22 options, 4.9 million configurations

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Comprex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes

Density Converter

22 options, 4.9 million configurations

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Comprex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes
Learning/Analysis Time	1.6 minutes	0.2 seconds	8.5 minutes

Density Converter

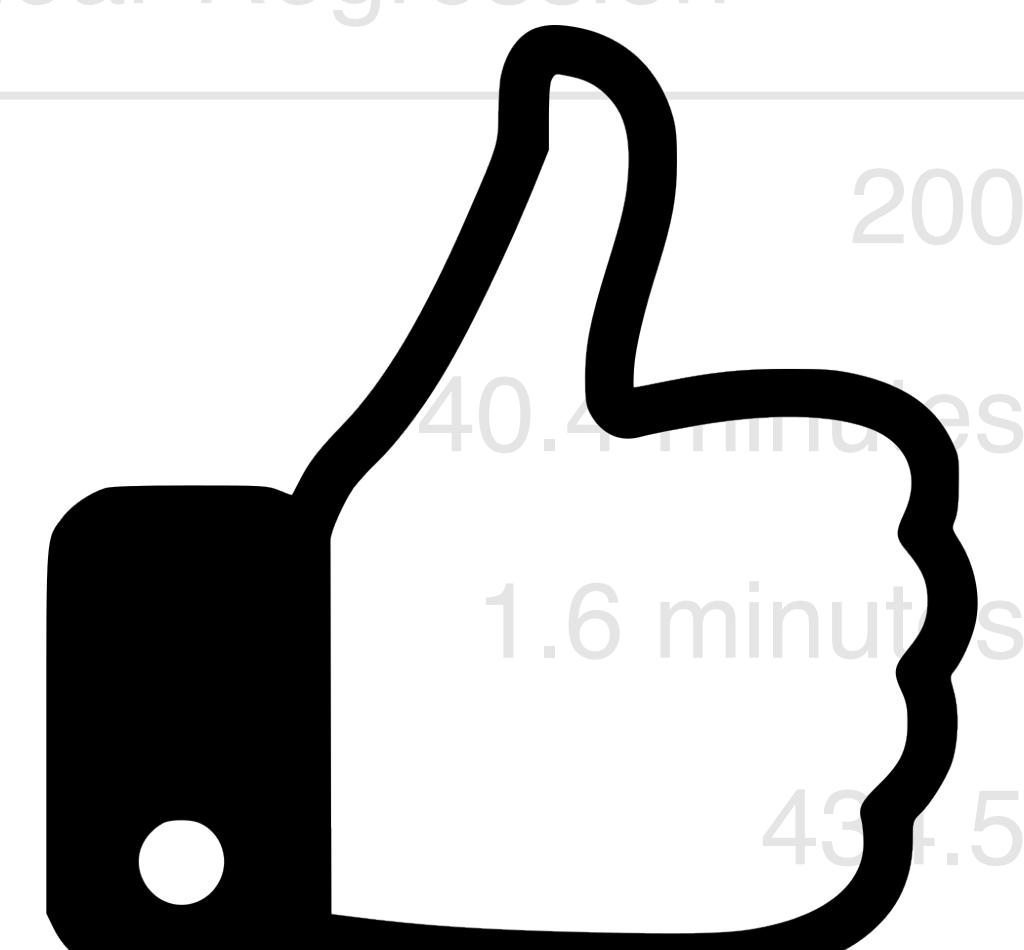
22 options, 4.9 million configurations

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Comprex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes
Learning/Analysis Time	1.6 minutes	0.2 seconds	8.5 minutes
MAPE (lower is better)	434.5	5.5	9.4

Density Converter

Efficient and accurate predictions

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Complex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes
Learning/Analysis Time	1.6 minutes	0.2 seconds	8.5 minutes
MAPE (lower is better)	434.5	5.5	9.4



Density Converter

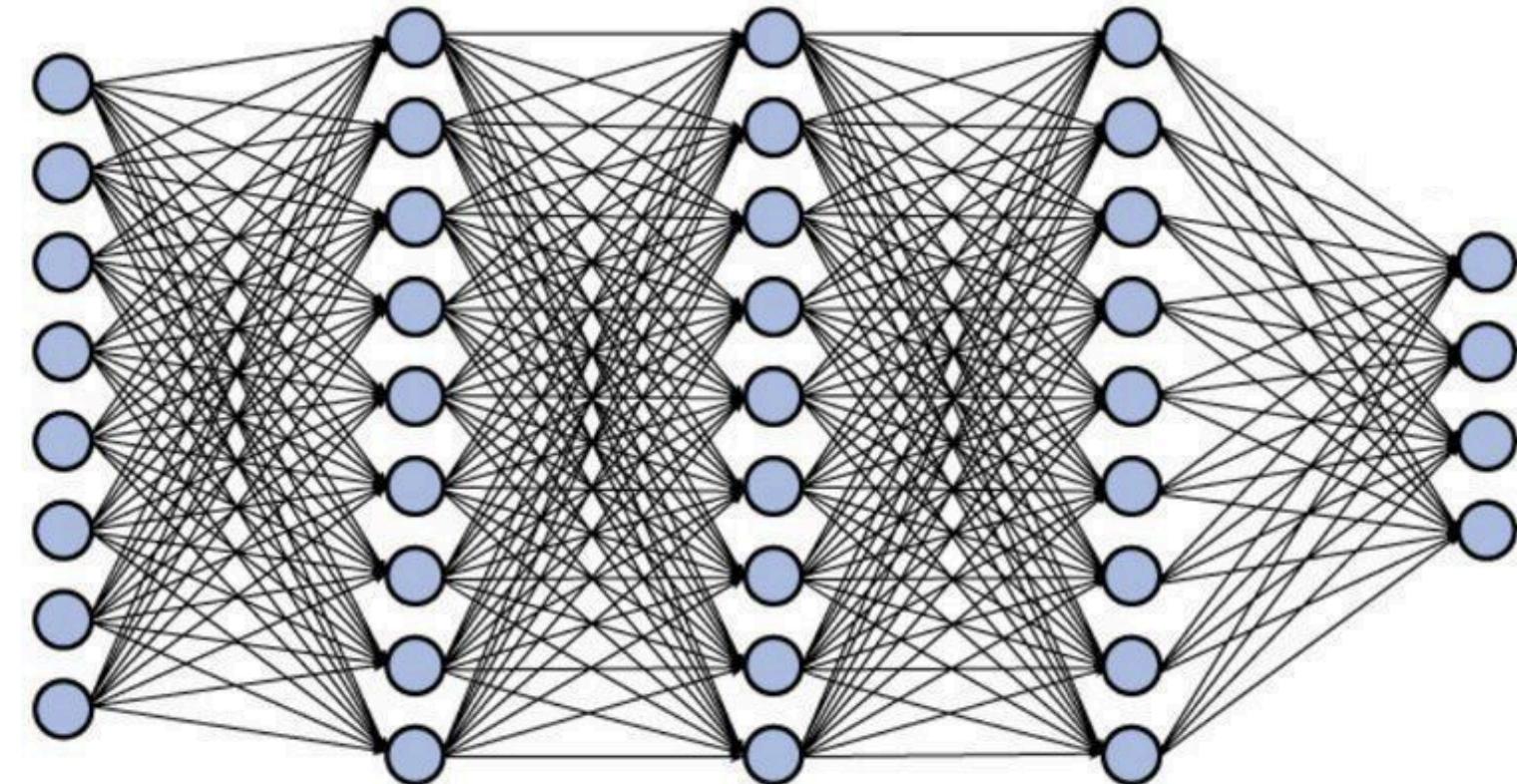
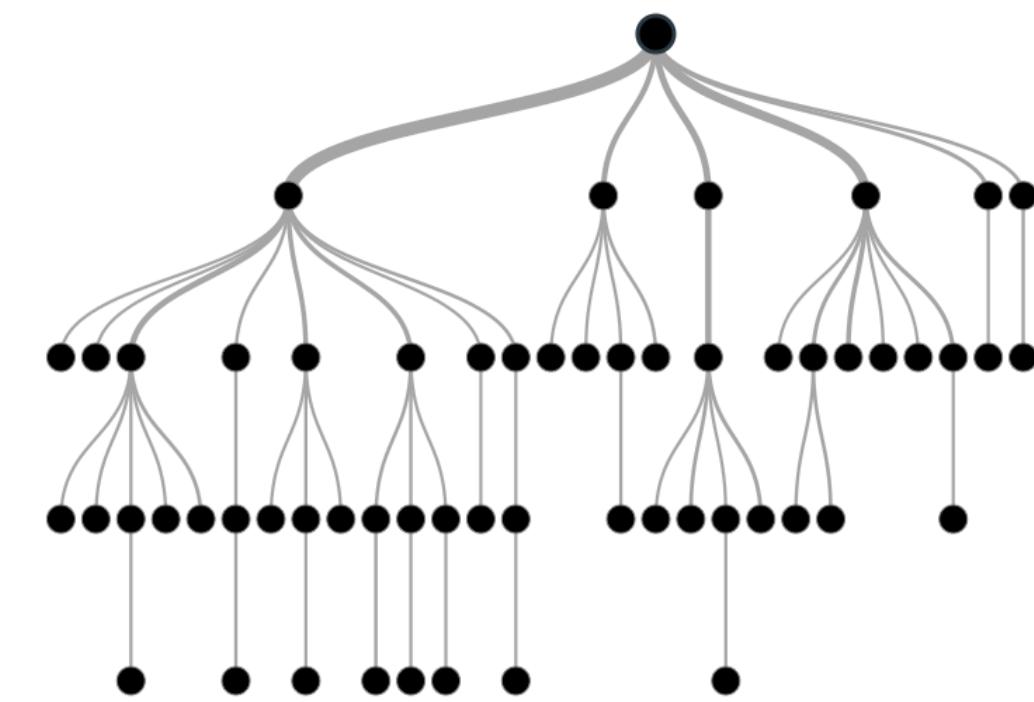
22 options, 4.9 million configurations

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Comprex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes
Learning/Analysis Time	1.6 minutes	0.2 seconds	8.5 minutes
MAPE (lower is better)	434.5	5.5	9.4
Interpretability	Easy	Difficult	Easy

Inherently interpretable

$$\begin{aligned} T = & 25 \\ & + 3 \cdot \text{Logging} \\ & - 5 \cdot \text{Indexing} \\ & + 9 \cdot \text{Compression} \cdot \text{Encryption} \end{aligned}$$

Not inherently interpretable



Density Converter

Efficient and accurate predictions

Understanding and debugging



Density Converter

Efficiently build accurate performance-influence models

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Complex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes
Learning/Analysis Time	1.6 minutes	0.2 seconds	8.5 minutes
MAPE (lower is better)	434.5	5.5	9.4
Interpretability	Easy	Difficult	Easy

Density Converter

Efficiently build accurate performance-influence models

Metric	200 Random & Stepwise Linear Regression	200 Random & Random Forest	Complex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes
Training/Testing Time	1.6 minutes	0.2 seconds	8.5 minutes
MAPE (lower is better)	434.5	5.5	9.4
Interpretability	Easy	Difficult	Easy

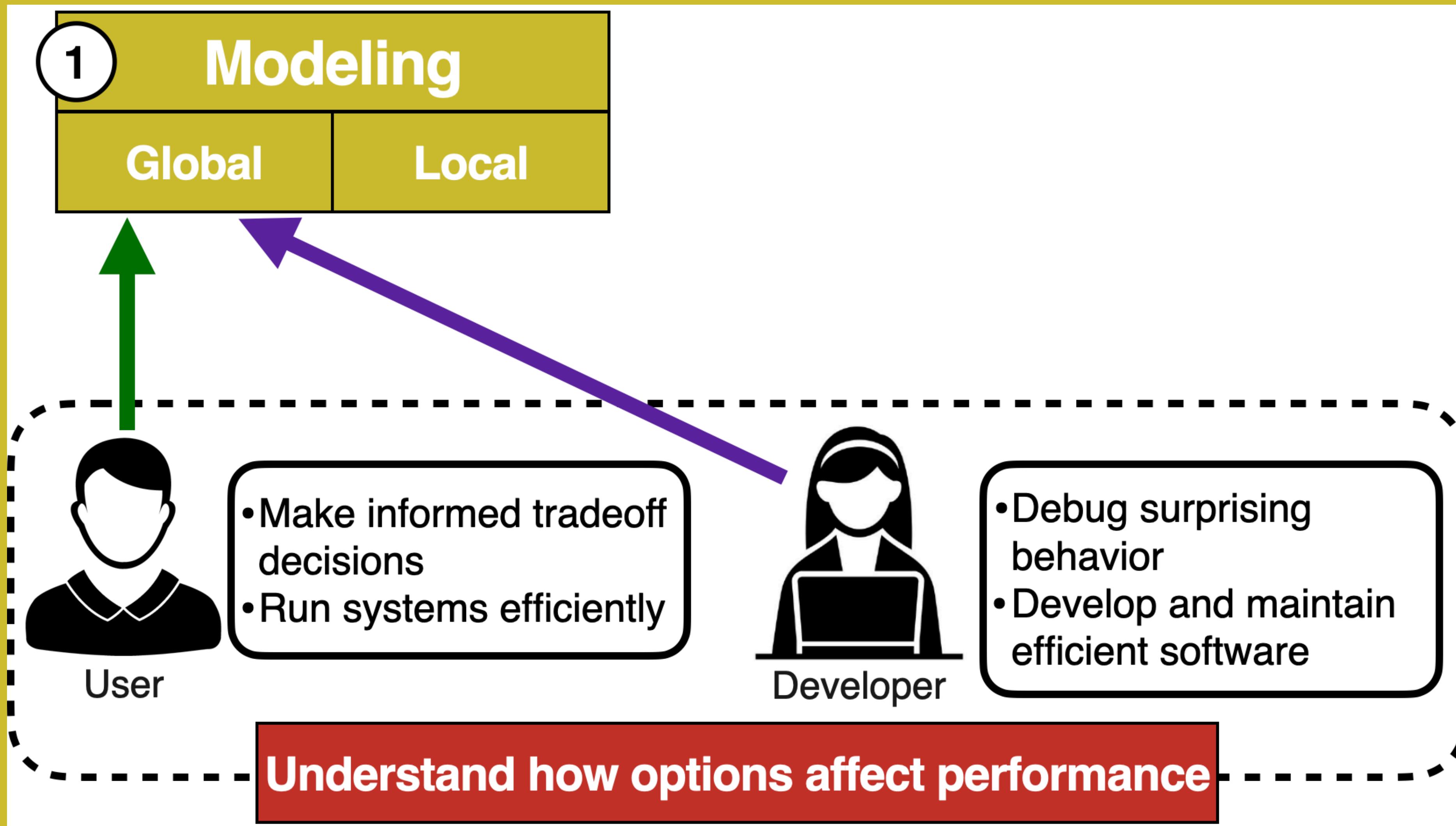
Density Converter Proposed Work

22 options, 4.9 million configurations

Analyze all subject systems with both prototypes

Metric	200 Random & Stepwise	200 Random & Random Forest	Comrex
# Configurations	200	200	88
Measuring time	40.4 minutes	40.4 minutes	16.6 minutes
Learning/Analysis Time	1.6 minutes	0.2 seconds	8.5 minutes
MAPE (lower is better)	434.5	5.5	9.4
Interpretability	Easy	Difficult	Easy

White-box Performance Modeling of Configurable Systems



Thesis Statement

White-box analysis of how options influence the performance of code-level structures in configurable systems:

- (1) helps to **efficiently build accurate and interpretable global and local performance-influence models**
- (2) guides developers to **inspect, understand, and debug configuration-related performance behaviors.**

$T = 25$

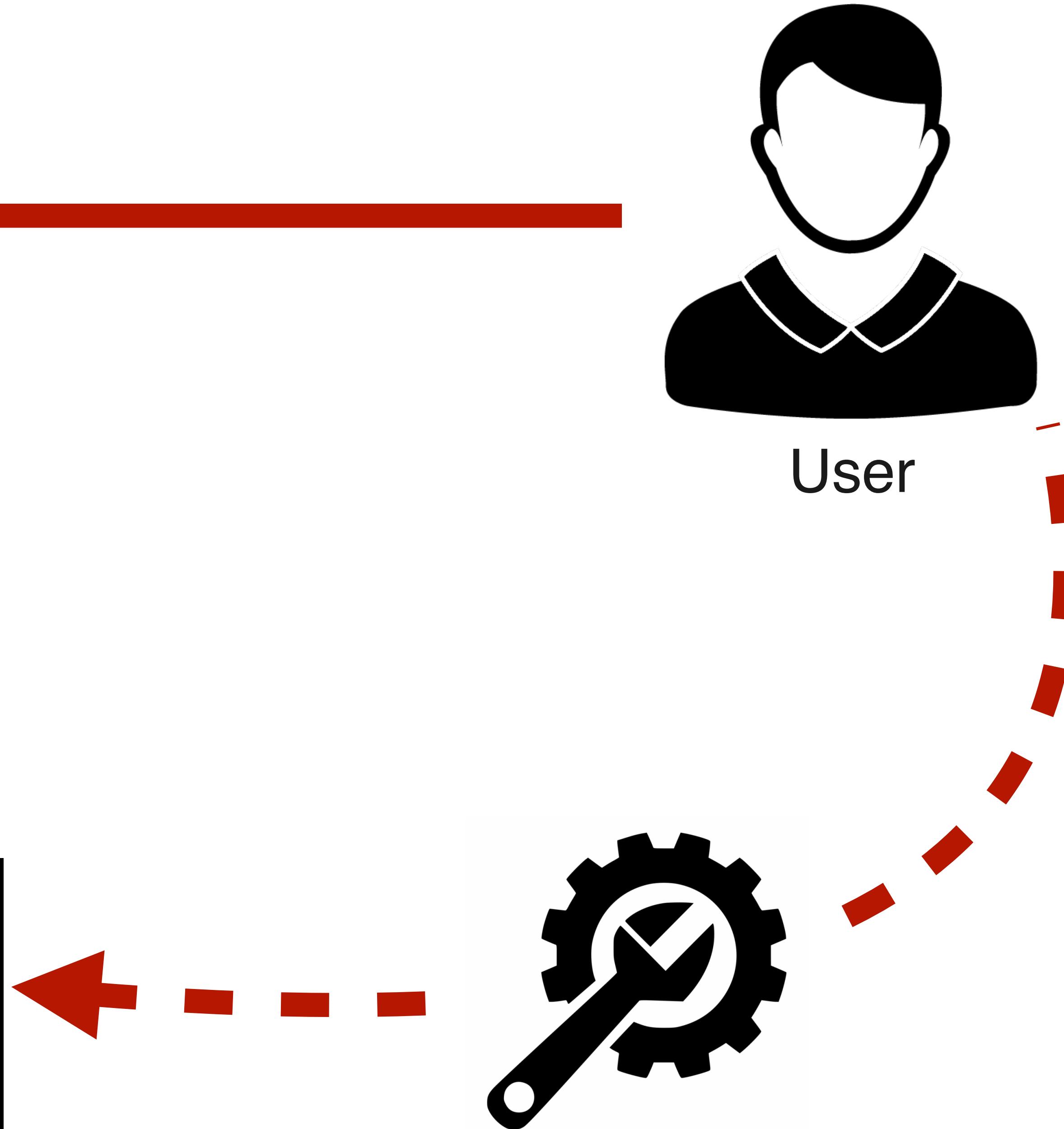
- + 3 · Logging
- 5 · Indexing
- + 9 · Compression · Encryption



User

System

$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption

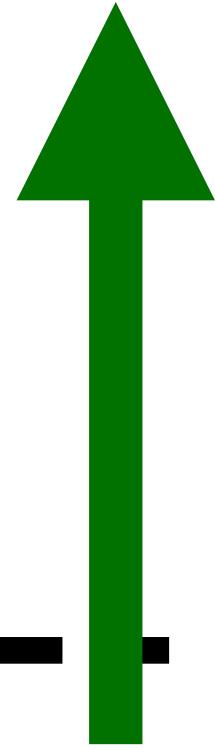


1

Modeling

Global

Local



User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

$T = 25$

- + 3 · Logging
- 5 · Indexing
- + 9 · Compression · Encryption

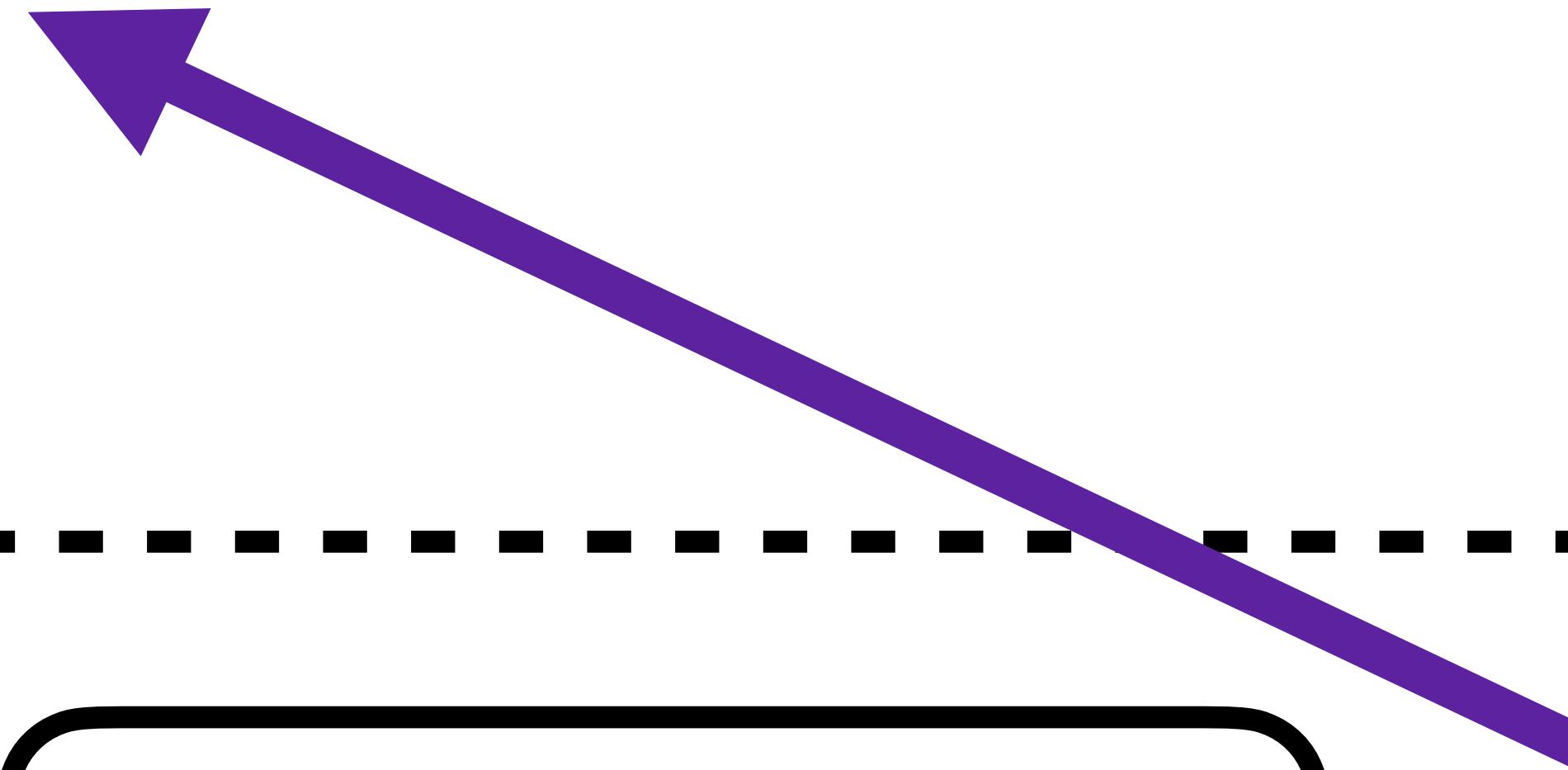
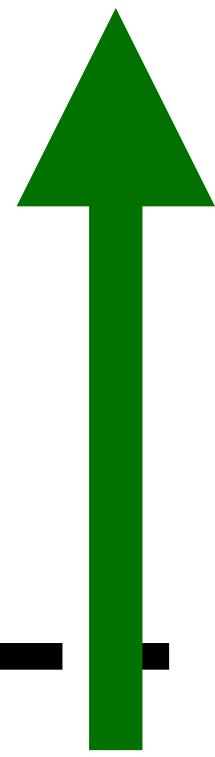


1

Modeling

Global

Local



User

- Make informed tradeoff decisions
- Run systems efficiently



Developer

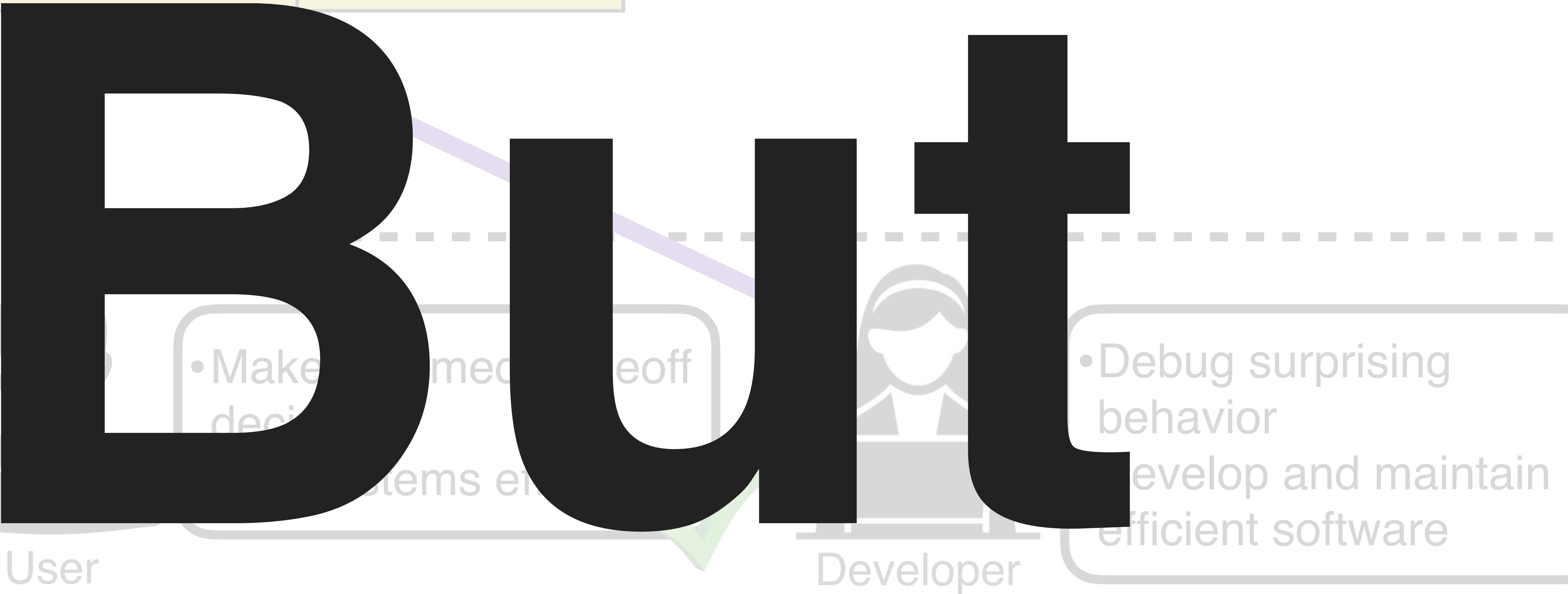
- Debug surprising behavior
- Develop and maintain efficient software

Understand how options affect performance

Modeling

Global

Local



Understand how options affect performance

$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption



System

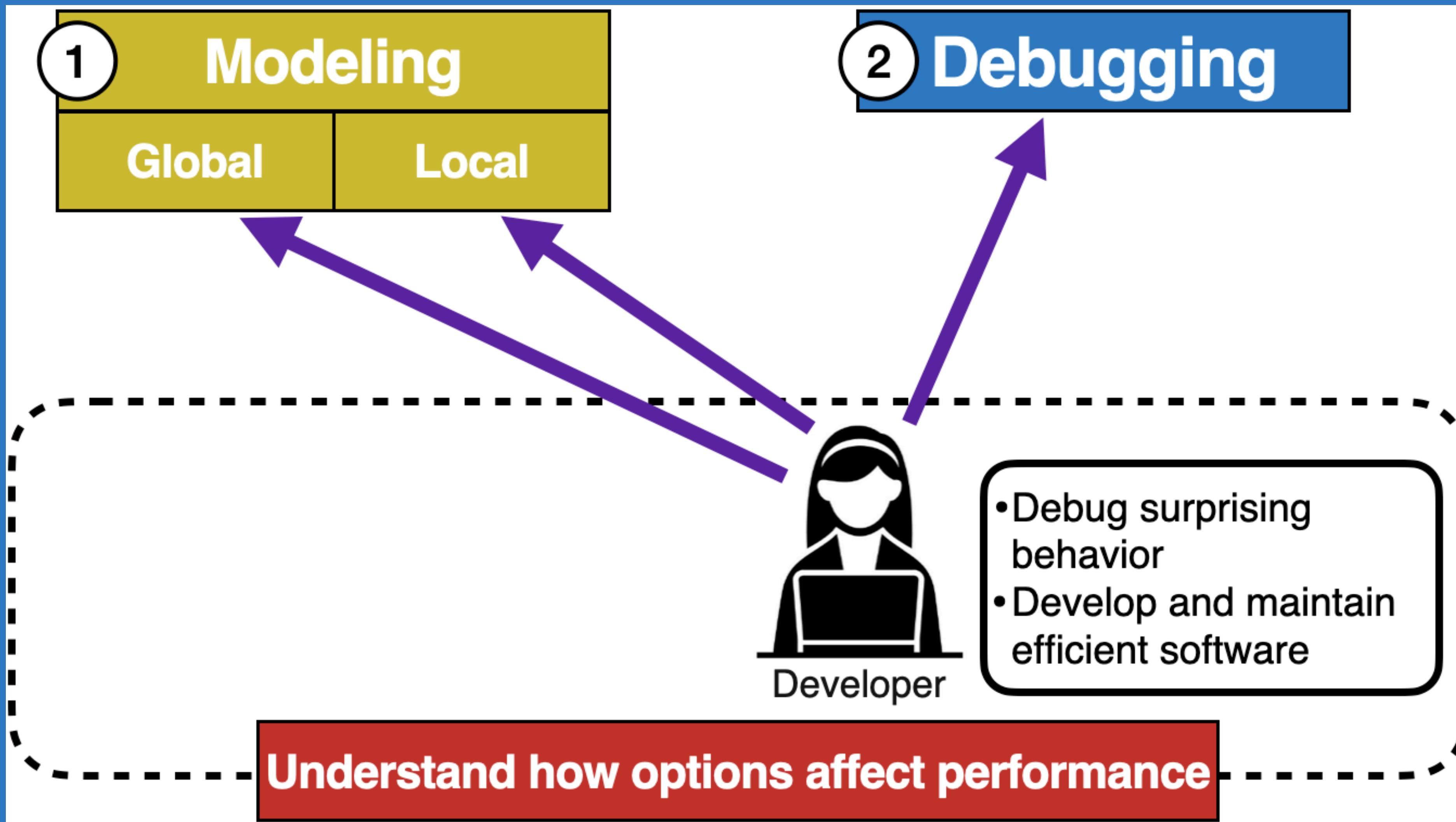
$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption

System



Developer

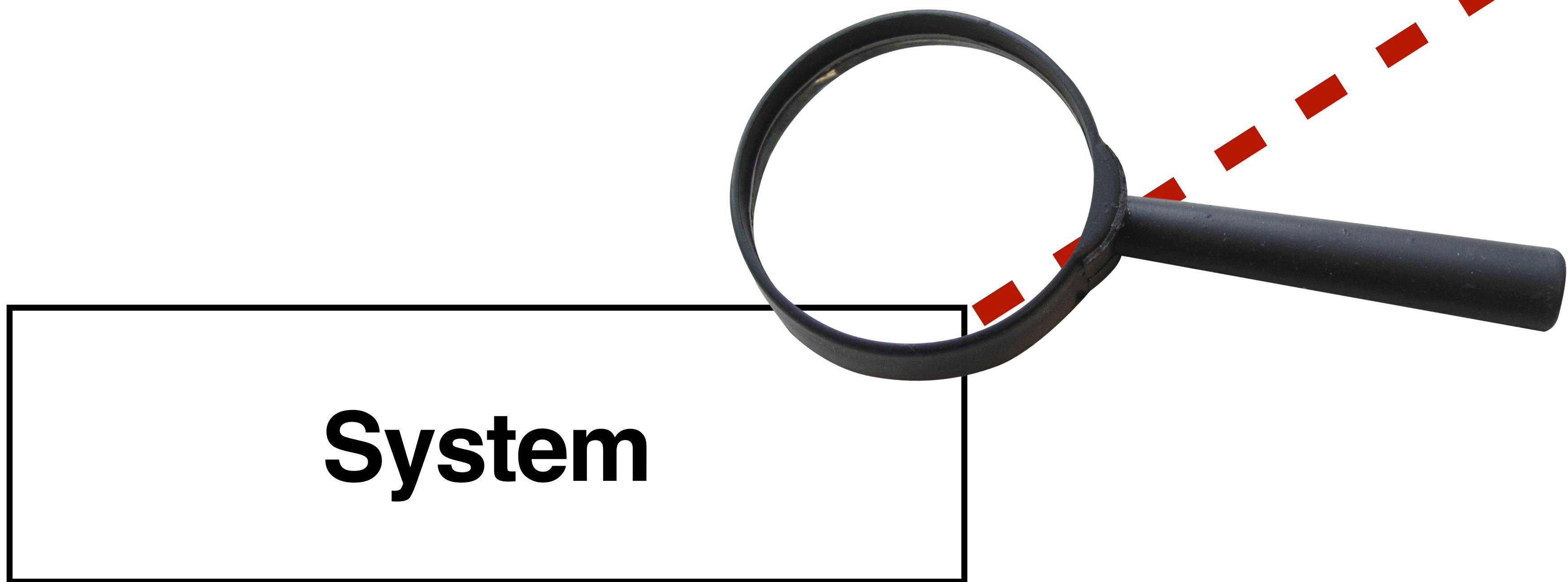
White-box Performance Debugging in Configurable Systems



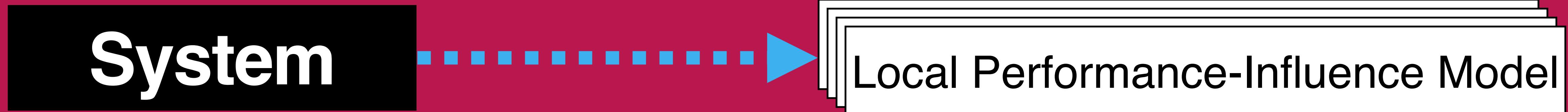
$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption



Developer



Recall Compositionality



Local performance-influence models explain how options and their interactions affect the performance of regions

Exploratory Study of Local Performance-Influence Models

Goal:

Investigate the usefulness of local performance-influence models

Understand how options affect the performance of a system in the implementation

Exploratory Study of Local Performance-Influence Models

Symptoms vs causes models

Understand how options affect the performance of a system in
the implementation

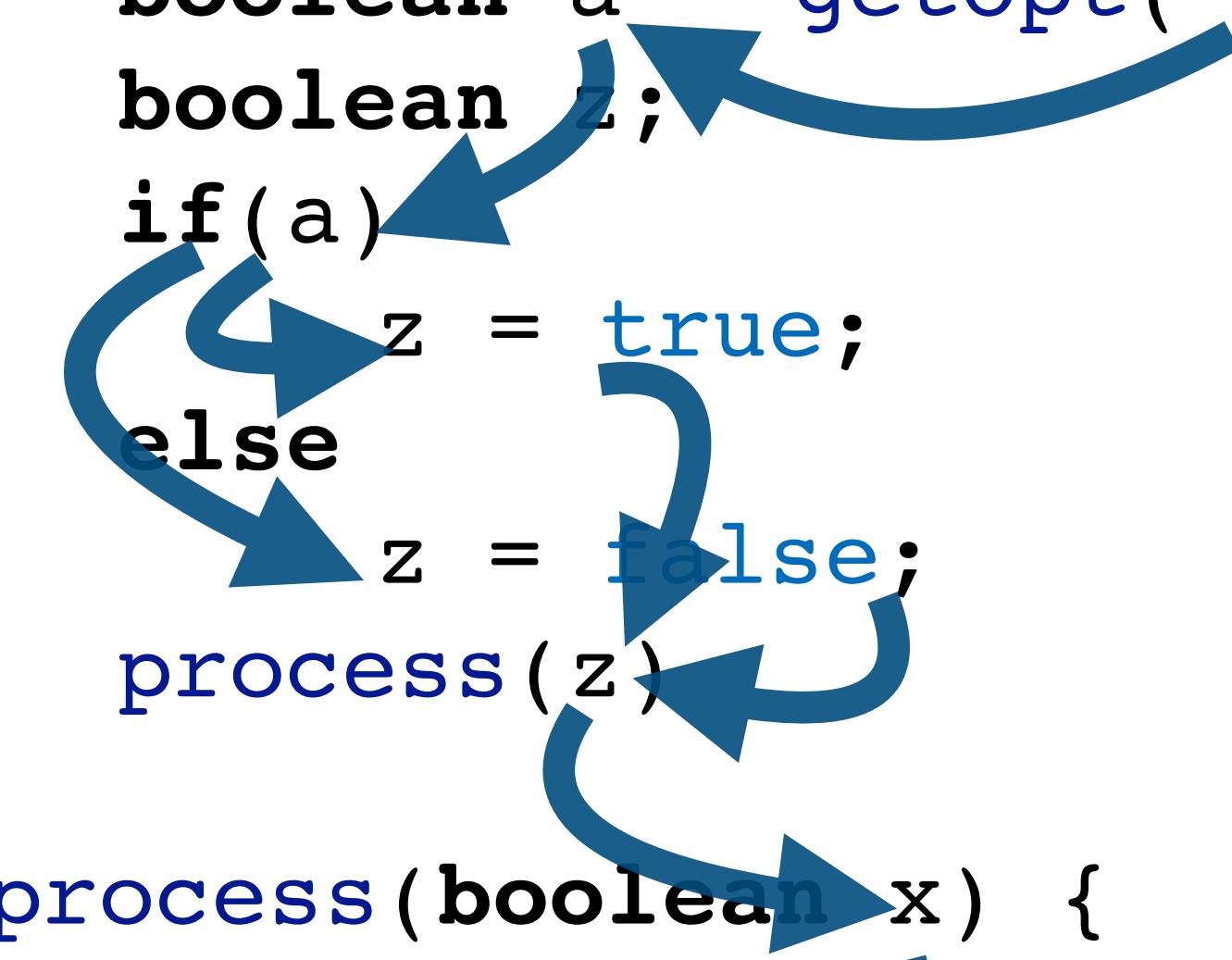
Symptom

```
1 def main() {  
2     boolean a = getOpt("Caching");  
3     boolean z;  
4     if(a)  
5         z = true;  
6     else  
7         z = false;  
8     process(z)  
9 }  
10 def process(boolean x) {  
11     if(x)  
12         ... // 3 seconds  
13     else  
14         ... // 2 seconds  
15 }
```

$$T_{\text{process}} = 2 + 1 \cdot \text{Caching}$$

Causes

```
1 def main() {  
2     boolean a = getOpt("Caching");  
3     boolean z;  
4     if(a)  
5         z = true;  
6     else  
7         z = false;  
8     process(z)  
9 }  
10 def process(boolean x) {  
11     if(x)  
12         ... // 3 seconds  
13     else  
14         ... // 2 seconds  
15 }
```



Symptom

```
1 def main() {  
2     boolean a = getOpt("Caching");  
3     boolean b;  
4     if(a)  
5         b = true;  
6     else  
7         z = false;  
8     process(z);  
9 }  
10 def process(boolean x) {  
11     if(x)  
12         ... // 1 second  
13     else  
14         ... // 2 seconds  
15 }
```

Still need additional information and help

Causes

```
1 def main() {  
2     boolean a = getOpt("Caching");  
3     boolean b;  
4     if(a)  
5         b = true;  
6     else  
7         z = false;  
8     process(z);  
9 }  
10 def process(boolean x) {  
11     if(x)  
12         ... // 1 second  
13     else  
14         ... // 2 seconds  
15 }
```

$$T_{process} = 2 + 1 \cdot Caching$$

Design Tool Support

Goal: Design and develop debugging tool support

Two-Part Exploratory Study

Research Question: What is the process and information needs when debugging the performance of configurable systems?

Begin debugging performance

Study 1.1

Goal: Are global and local performance-influence models helpful?

Debug Performance of Hotspot Regions

Study 1.2

Later in the debugging process
Identified options and locations of interest

Goal: Guide design of new tool support



**Conducted studies with 14 researchers/developers working
in academia**

**Expect to conduct studies with 5 developers working in
industry**

Preliminary Results - Study 1.1 - Process

1. Which options affect performance the most?

Hypothesis: Global performance-influence models

Conducted studies with 14 researchers/developers working
in academia $T = 25$

+ 3 · Logging

- 5 · Indexing

+ 9 · Compression · Encryption

Expect to conduct studies with 5 developers working in
industry

Preliminary Results - Study 1.1 - Process

2. Where do options affect performance the most?

Hypothesis: Local performance-influence models

Conducted studies with 14 researchers/developers working
in academia

$$T_{\text{main}} = 2$$

$$T_{\text{process}} = 2 \cdot \text{Compression} \cdot \text{Encryption}$$

Expect to conduct studies with 5 developers working in
industry

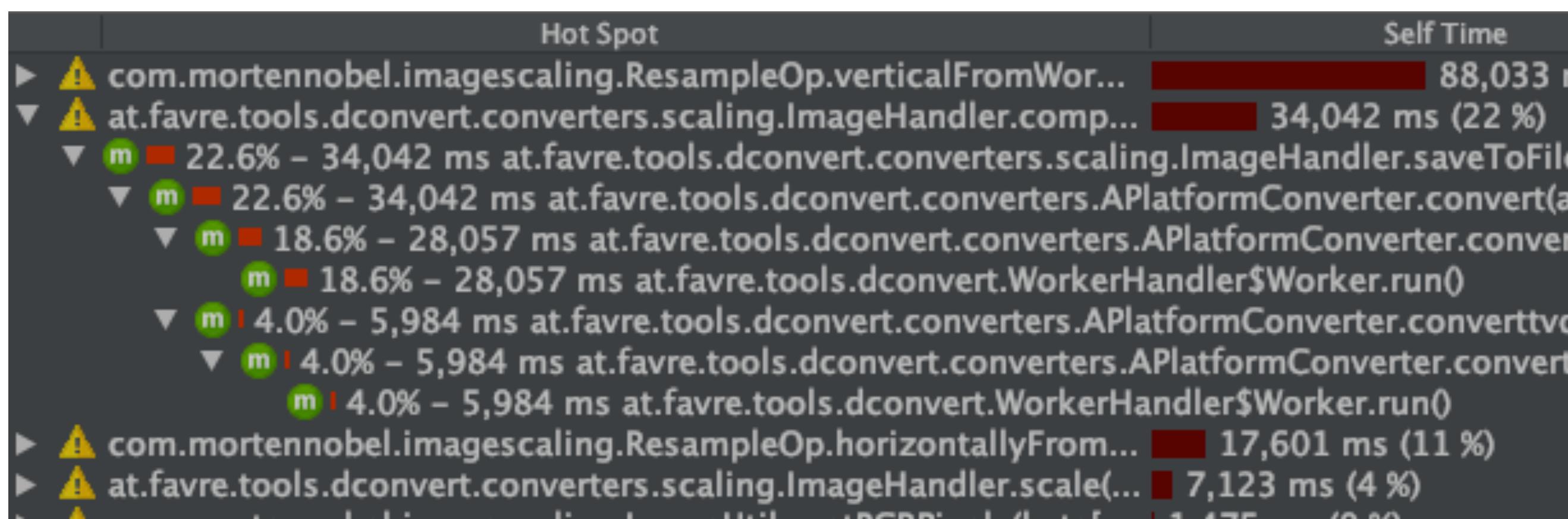
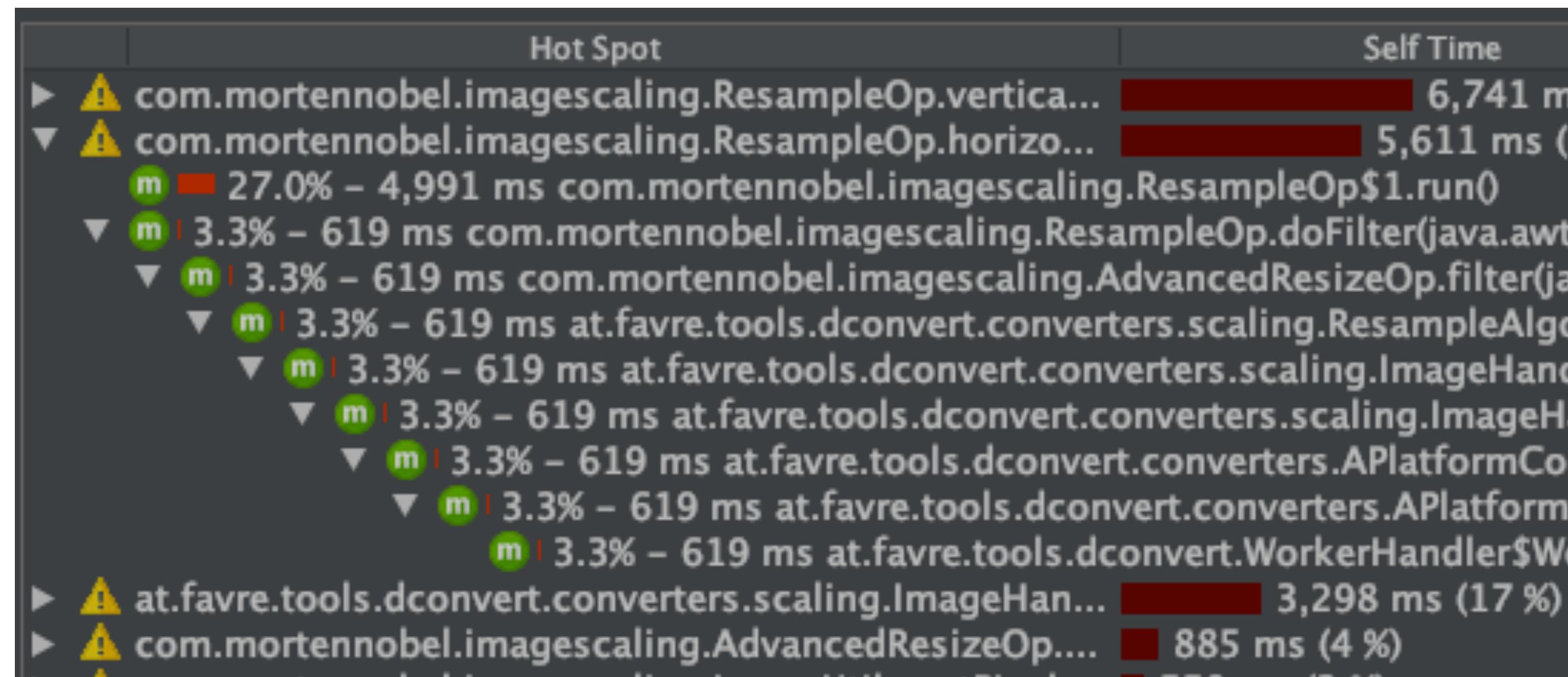
$$T_{\text{convert}} = 7 \cdot \text{Compression} \cdot \text{Encryption}$$

Preliminary Results - Study 1.2 - Information needs

Trace options

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean z;  
4     if(a)  
5         z = true;  
6     else  
7         z = false;  
8     process(z);  
9 }  
10 def process(boolean x) {  
11     if(x)  
12         ... // 3 seconds  
13     else  
14         ... // 2 seconds  
15 }
```

Compare performance profiles



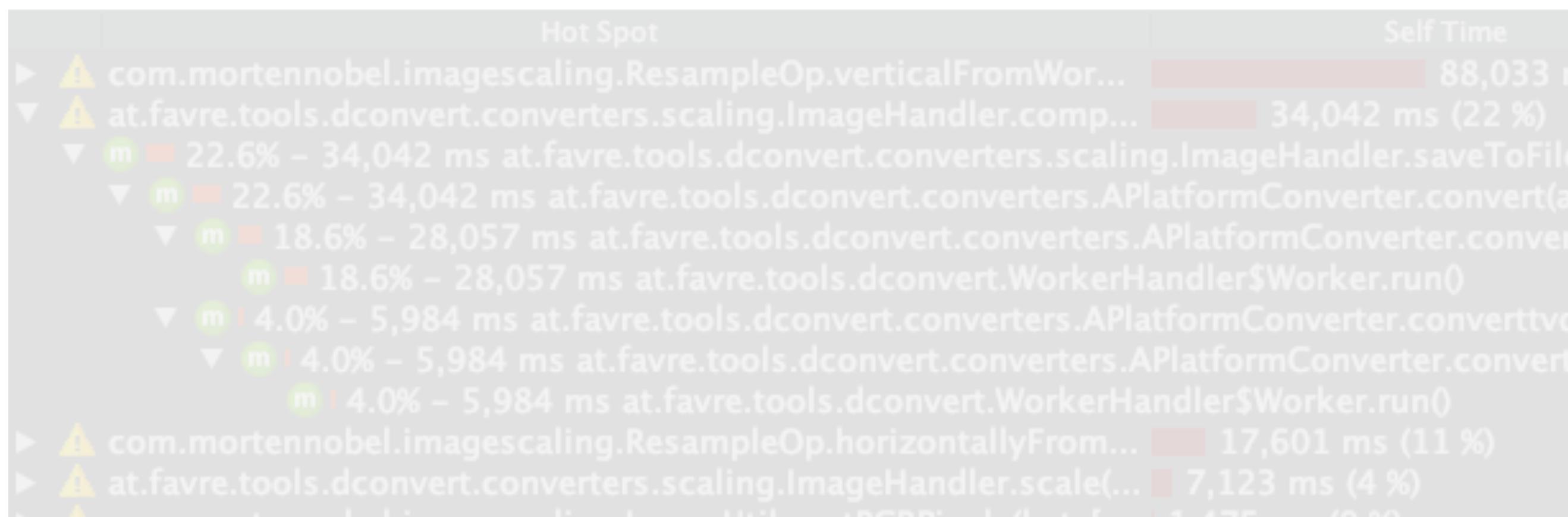
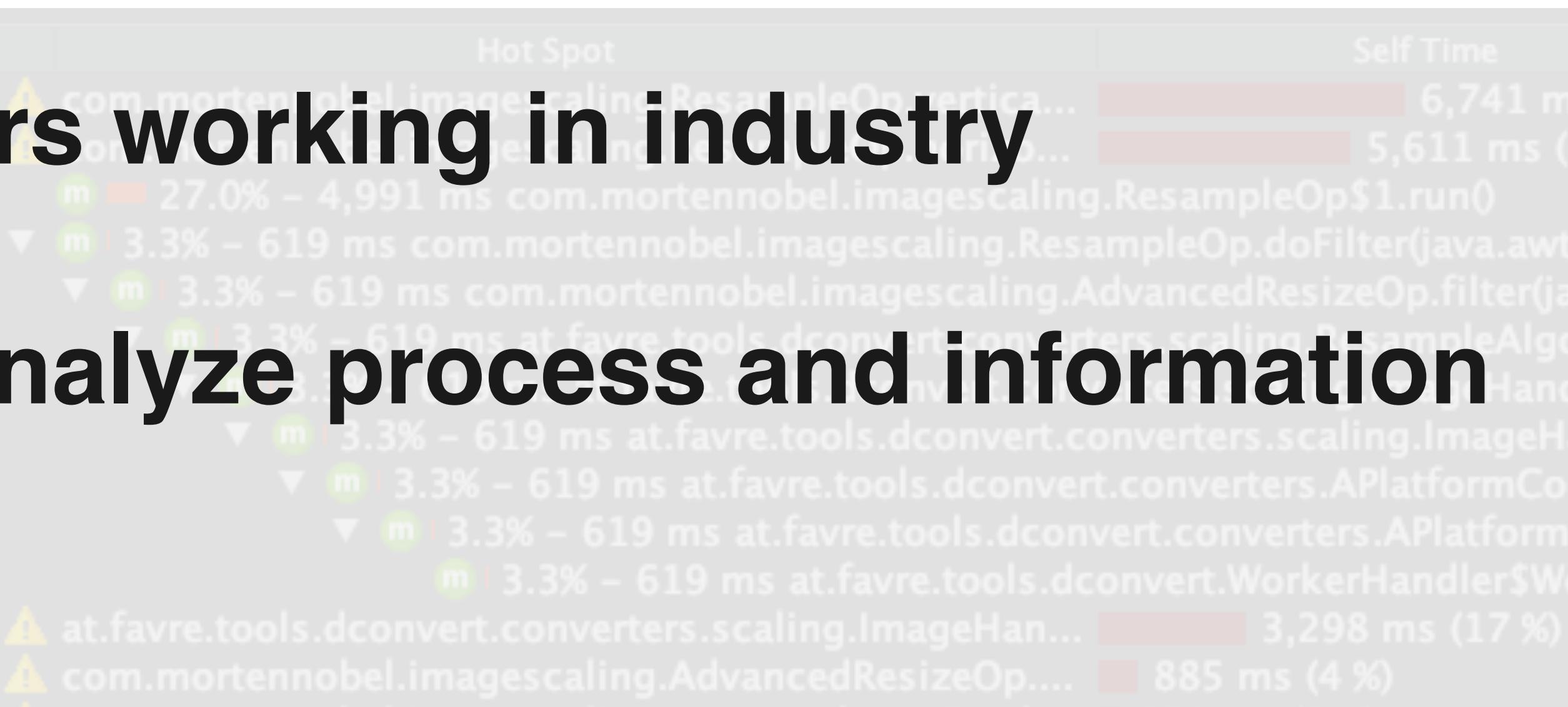
Preliminary Results - Study 1.2 - Information needs Proposed Work

Trace options

```
1 def main() {  
2     boolean a;  
3     if(a)  
4         z = false;  
5     else  
6         process(z);  
7 }  
8  
9 def process(boolean x) {  
10    if(x)  
11        ... // 3 seconds  
12    else  
13        ... // 2 seconds  
14 }  
15 }
```

Conduct studies with 5 developers working in industry

Quantitatively and qualitatively analyze process and information needs



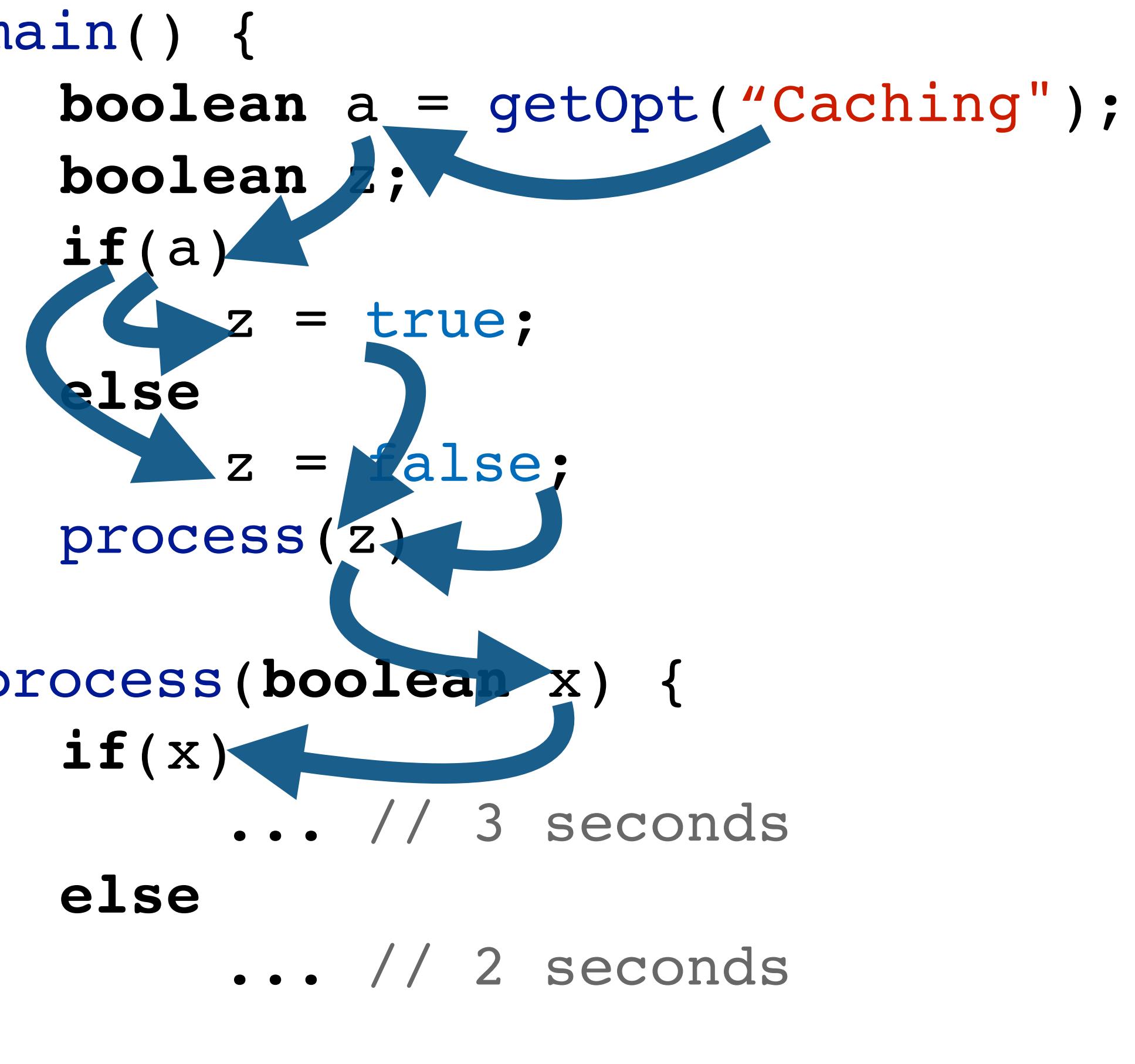
Design Tool Support

Goal: Design and develop debugging tool support

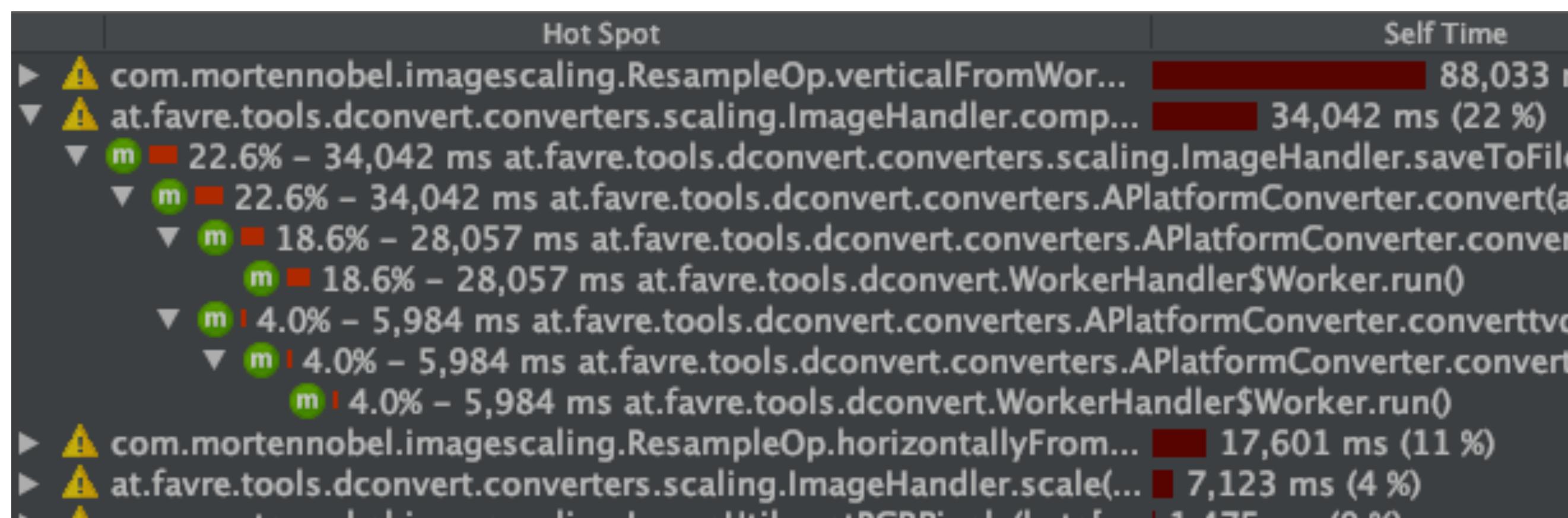
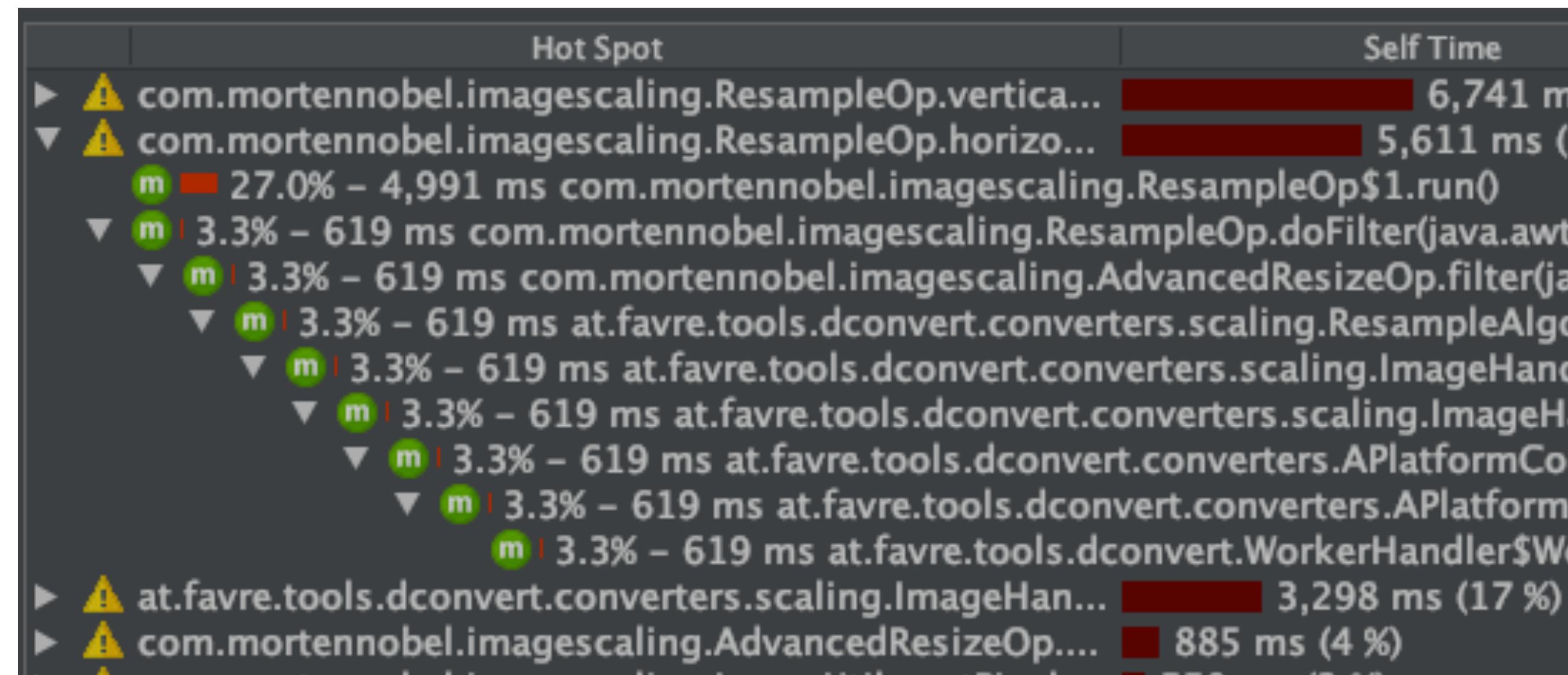
Examples of Tool Support

Trace options

```
1 def main() {  
2     boolean a = getopt("Caching");  
3     boolean z;  
4     if(a)  
5         z = true;  
6     else  
7         z = false;  
8     process(z);  
9 }  
10 def process(boolean x) {  
11     if(x)  
12         ... // 3 seconds  
13     else  
14         ... // 2 seconds  
15 }
```



Compare performance profiles



Examples of Tool support Proposed Work

Trace options

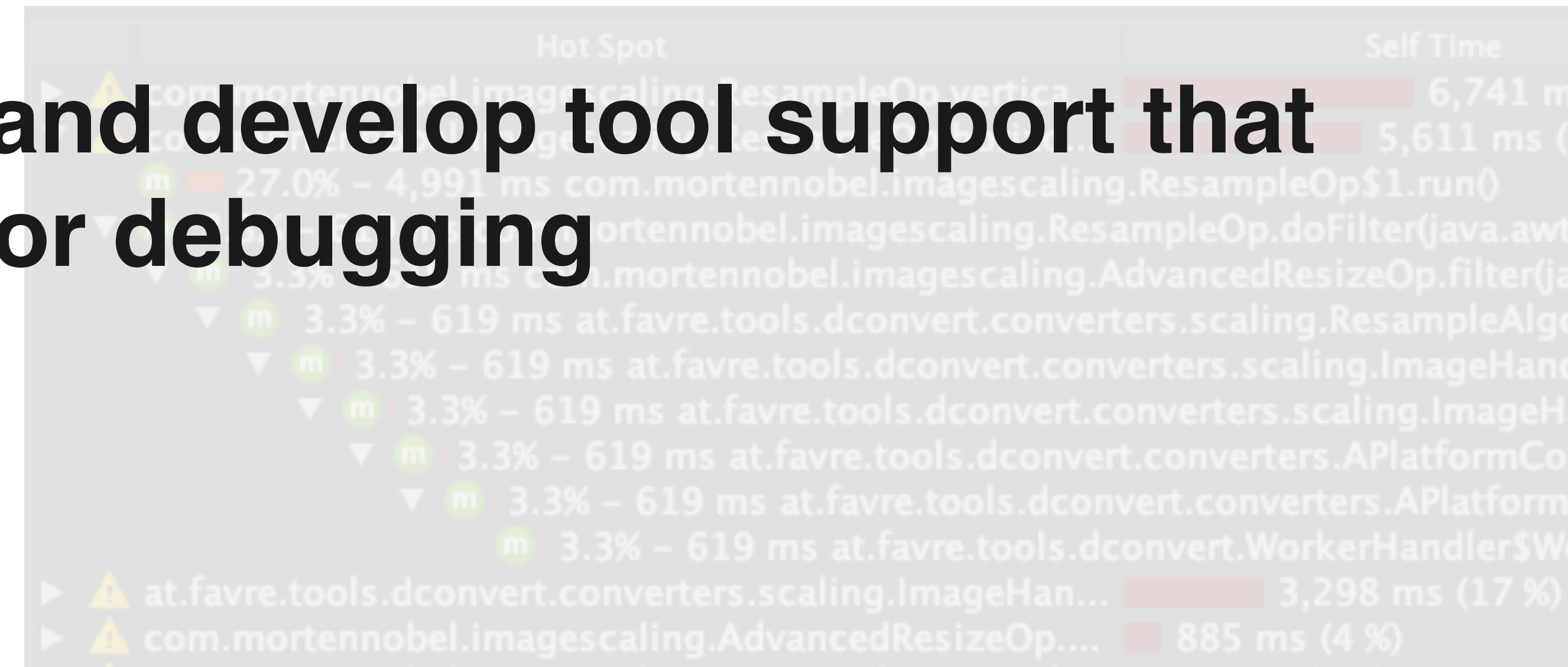
The diagram shows a Java code snippet with annotations and arrows indicating flow and variable state. The code is as follows:

```
1 def main() {  
2     boolean a = true;  
3     if (a) {  
4         z = true;  
5     } else {  
6         z = false;  
7     }  
8     process(z);  
9 }  
10 def process(boolean x) {  
11     if (x) {  
12         ... // 3 seconds  
13     } else {  
14         ... // 2 seconds  
15     }  
16 }
```

Annotations and arrows highlight the following points:

- A blue arrow points from line 5 to line 7, indicating the flow of control from the `if` block to the `else` block.
- A blue arrow points from line 7 to line 8, indicating the flow of control from the `else` block to the `process(z)` call.
- A blue arrow points from line 8 to line 11, indicating the flow of control from the `process(z)` call back to the `if (x)` block.
- A red arrow points from line 11 to line 12, indicating the flow of control from the `if (x)` block to the code block following it.
- A red arrow points from line 12 to line 14, indicating the flow of control from the code block following the first `if` block to the code block following the second `if` block.
- Text annotations "z = true;" and "z = false;" are placed near lines 5 and 7 respectively, with arrows pointing to the assignment statements.
- Text annotations "... // 3 seconds" and "... // 2 seconds" are placed near lines 12 and 14 respectively, with arrows pointing to the code blocks following the `if` statements.

Guided by our studies; design; and develop tool support that provides relevant information for debugging



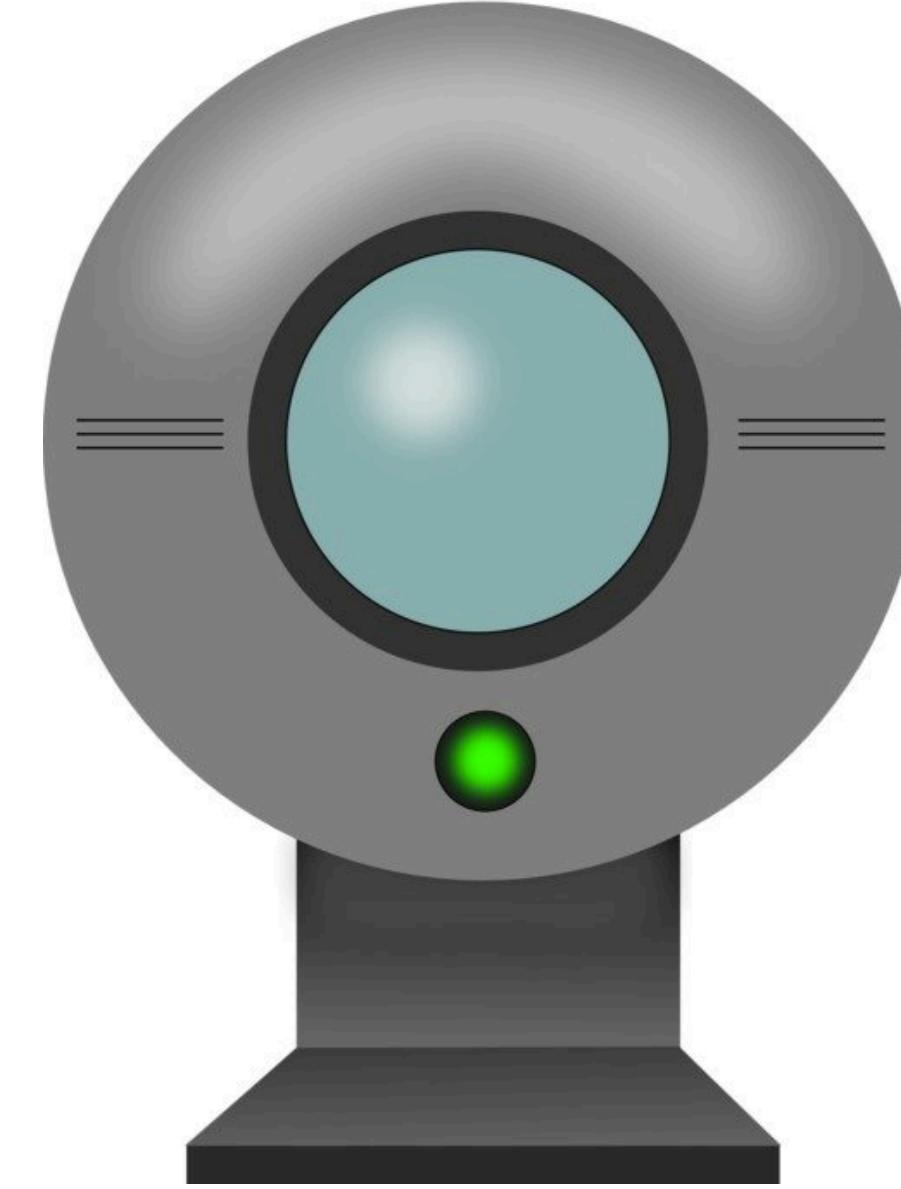
Validation Study

Study 2

Research Questions: To what extend are

- (1) Global performance-influence models
- (2) Local performance-influence models
- (3) Additional tool support

useful to debug the performance of configurable systems?



Expect to conduct studies with 10 researchers/developers working in academia

No control group: Participants cannot debug without tool support (Study 1.1 and Study 1.2)

Proposed Work

Conduct studies with 10 researchers/developers working in academia

Expect to conduct studies with 10 researchers/developers working in academia

No control group: Participants cannot debug without tool support (Study 1.1 and Study 1.2)

Optional Validation Study in the Field

Optional Study 3

Goal: Validate the usefulness of our tools in real bug reports and real systems

Option 1: Deploy tools in the field

Option 2: Collaborate with developers

Goal: Validate the usefulness of our tools in real bug reports and real systems

Option 3: Case studies

Option 1: Deploy tools in the field

Option 2: Collaborate with developers

Goal: Validate the usefulness of our tools in real bug reports and real systems

Option 3: Case studies

Option 1: Deploy tools in the field

Option 2: Collaborate with developers

Goal: Validate the usefulness of our tools in real bug reports
and real systems

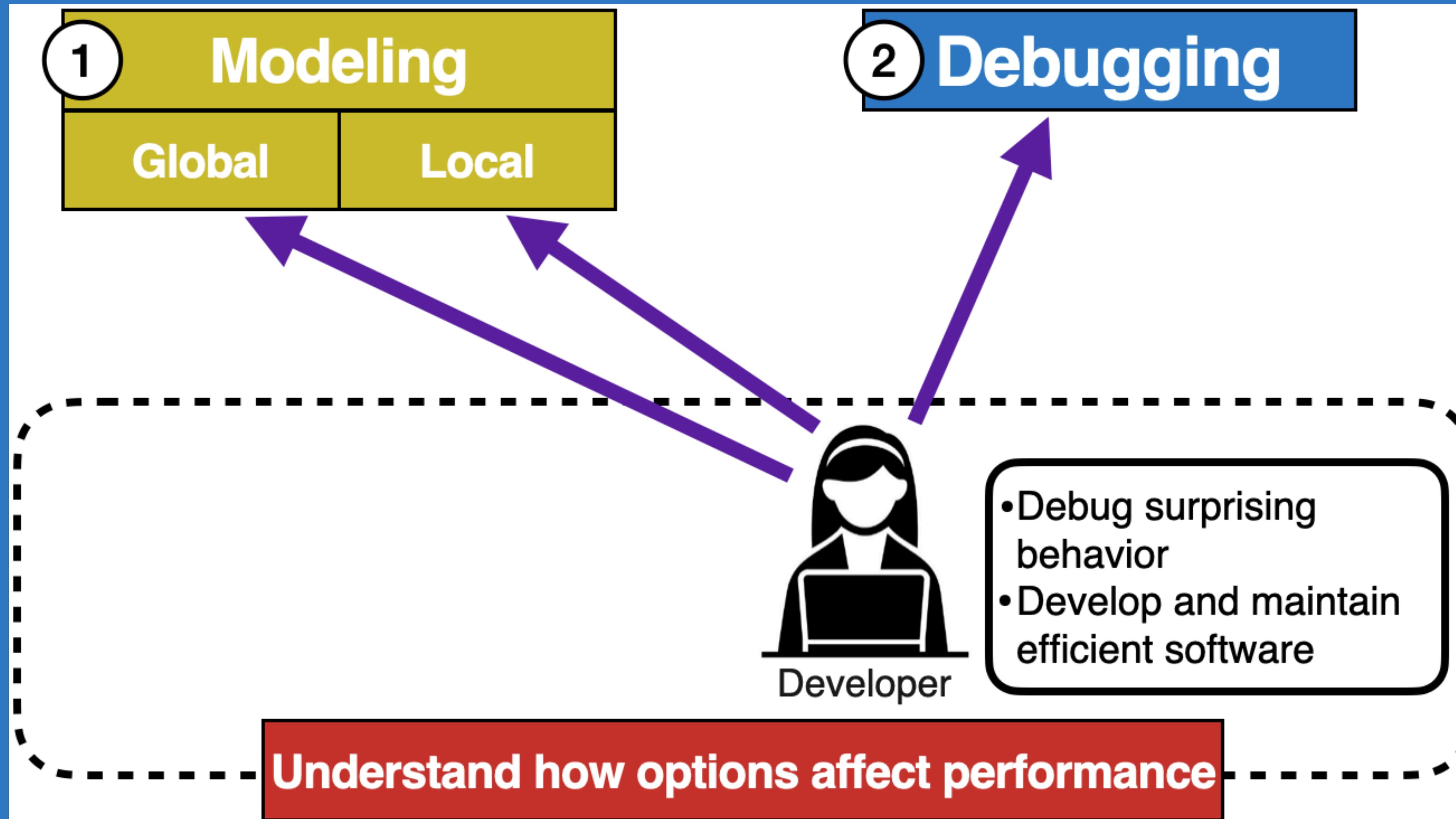
Option 3: Case studies

Optional Validation Study in the Field

Optional Study 3

Goal: Validate the usefulness of our tools in real bug reports and real systems

White-box Performance Debugging in Configurable Systems



Thesis Statement

White-box analysis of how options influence the performance of code-level structures in configurable systems:

(1) helps to **efficiently build accurate and interpretable global and local performance-influence models**

(2) guides developers to **inspect, understand, and debug configuration-related performance behaviors.**

$T = 25$
+ 3 · Logging
- 5 · Indexing
+ 9 · Compression · Encryption



Developer



System

$$T_{compress} = 5 \cdot \text{Indexing}$$

$$T_{process} = 9 \cdot \text{Compression} \cdot \text{Encryption}$$



1

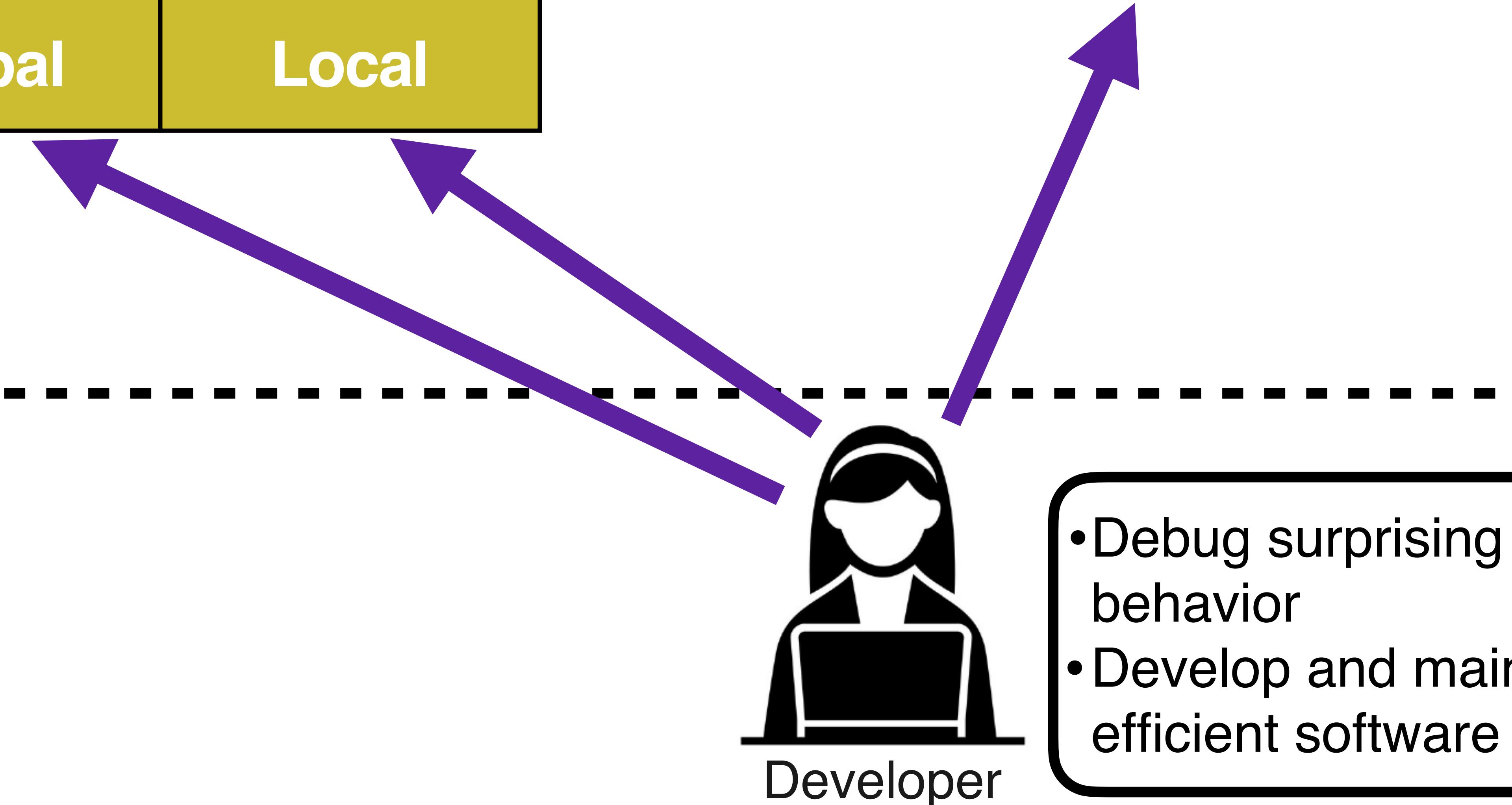
Modeling

Global

Local

2

Debugging



Understand how options affect performance

1

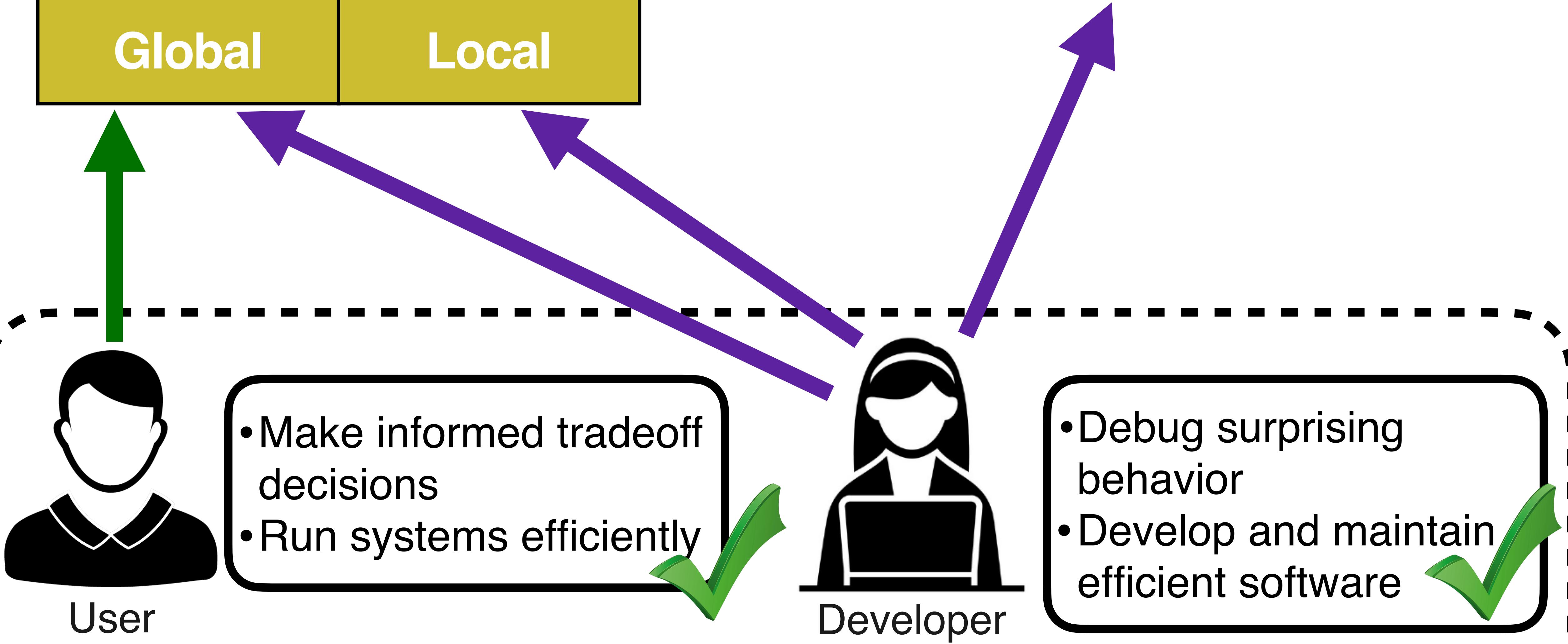
Modeling

Global

Local

2

Debugging



Understand how options affect performance

1

Modeling

Global

Local

2

Debugging

Research

- Make informed decisions
- Run systems efficiently

Plan

- Debug surprising behavior
- Develop and maintain efficient software

User

Developer

Understand how options affect performance

1

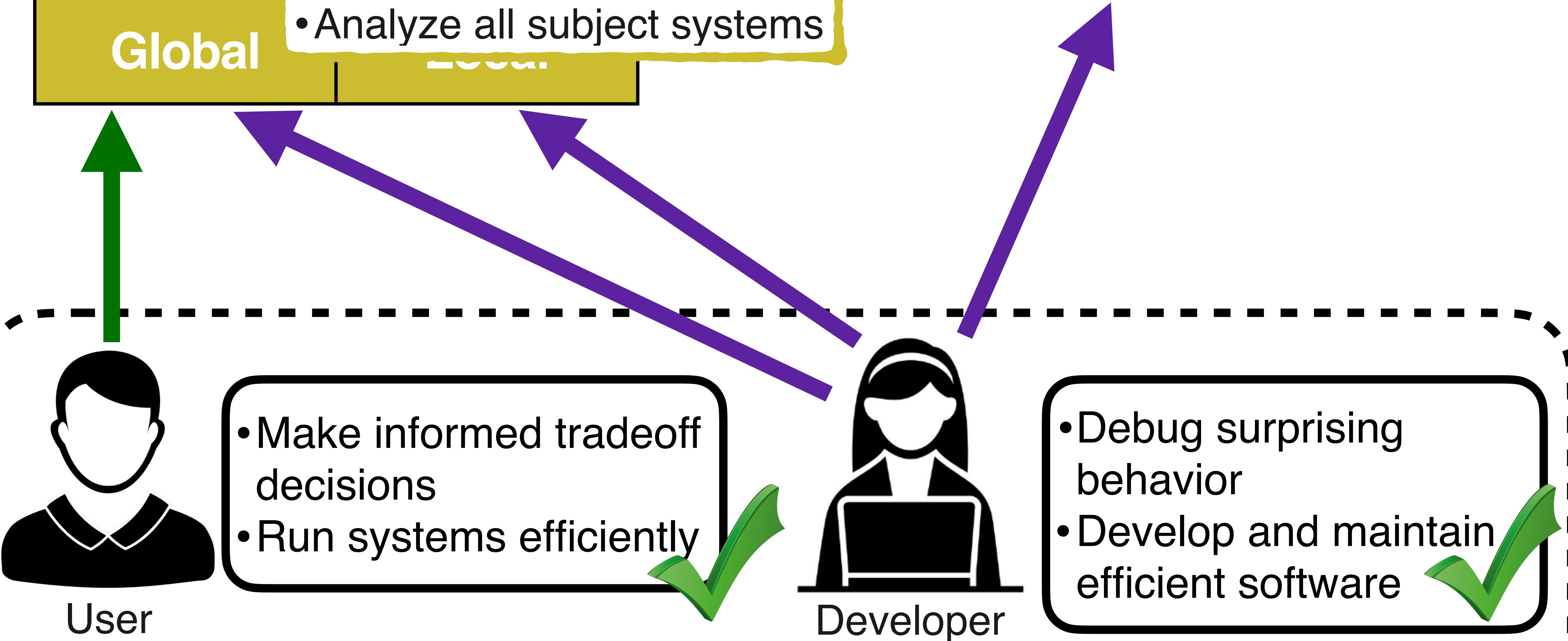
Mc

Global

- Polish Comprex
- Compare all approaches
- Analyze all subject systems

2

Debugging



Understand how options affect performance

1

Mc

Global

- Polish Comprex
- Compare all approaches
- Analyze all subject systems

2

Debugging

- Finish Studies 1.1 and 1.2
- Analyze Studies
- Implement new tool support
- Study 2
- Optional Study 3



User

- Make informed tradeoff decisions
- Run systems efficiently



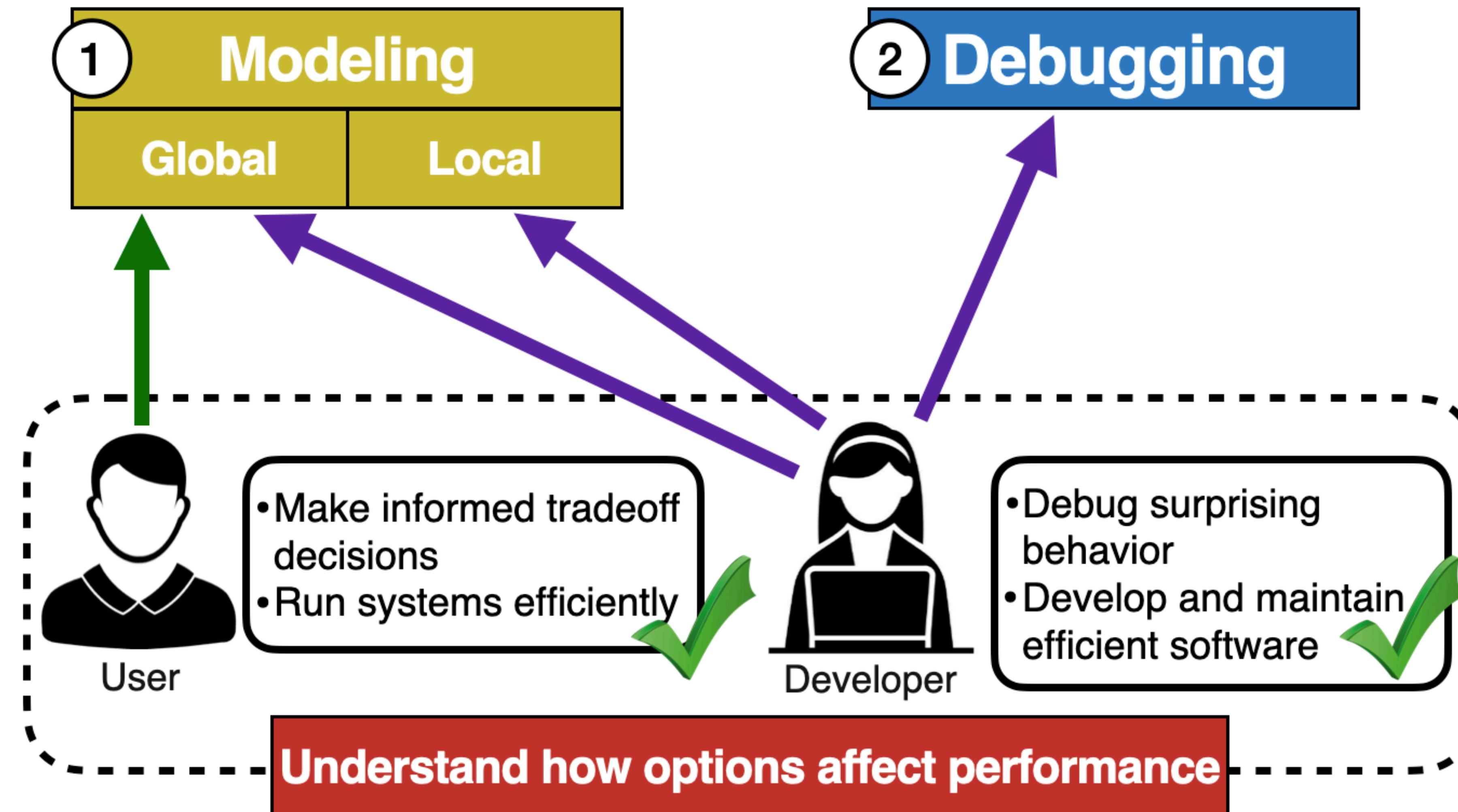
Developer

- Debug surprising behavior
- Develop and maintain efficient software



Understand how options affect performance

White-box Analysis for Modeling and Debugging the Performance of Configurable Systems



Extra slides

Sparse Linear Models

	Apache Lucene	H2	Berkeley DB	Density Converter
Terms	27	33	77	47
Regions	24	5	17	9

Modeling

Polish Comprex writing	2 weeks
Compare prototypes with all approaches	2 weeks
Analyze all subject systems with both prototypes	2 weeks

Debugging

Study 1.1 and Study 1.2 (Exploration)	2 months
Design and implement new tool support	6 months
Study 2 (Validation)	2 months
Optional Study 3 (Validation in the field)	2 months
Writing	1 month

Thesis writing and defense

2 months

Implicit flows

Region Analysis

Preliminary Results - Study 1.1 - Process

