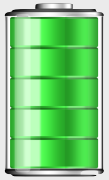
SEEDS: A Software Engineer's Energy Optimization Decision Support Framework.

Irene L. Manotas Gutiérrez

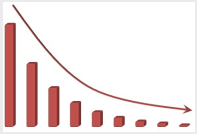
University of Delaware
November, 2013

Software Energy Usage: Why Care?

Reduced energy consumption of applications could:



Extend battery life (e.g., for mobile devices)



Reduce power electricity bills and related costs (e.g., cooling systems in data centers)

Support sustainability
(e.g., by software engineering sustainability)

System and Developer Roles

Historically, software developers have left concerns about power consumption to systems engineers.

Hardware
cpu, disk, etc.

Operating
Systems

Compilers

While this strategy has been successful, it is likely that encouraging software developers to participate in the process can result in more efficient applications.

Making Energy-related Decisions

```
Collection x = new ???
```

Which implementation is the best option for x?

LinkedHashSet
TreeSet
HashSet
EnumSet
CopyOnWriteArraySet

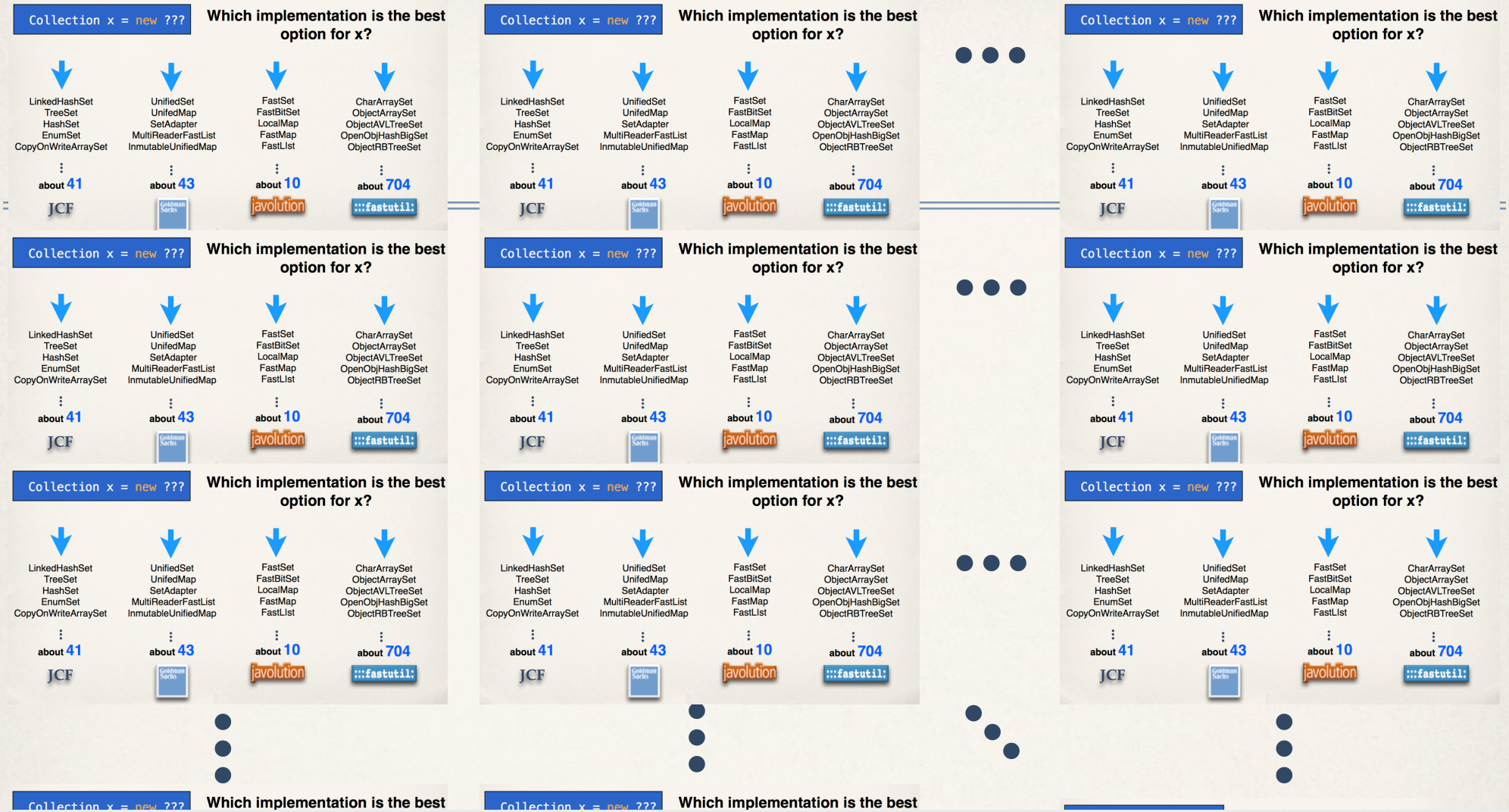
⋮

about 41

JCF

Hundreds of possible implementations

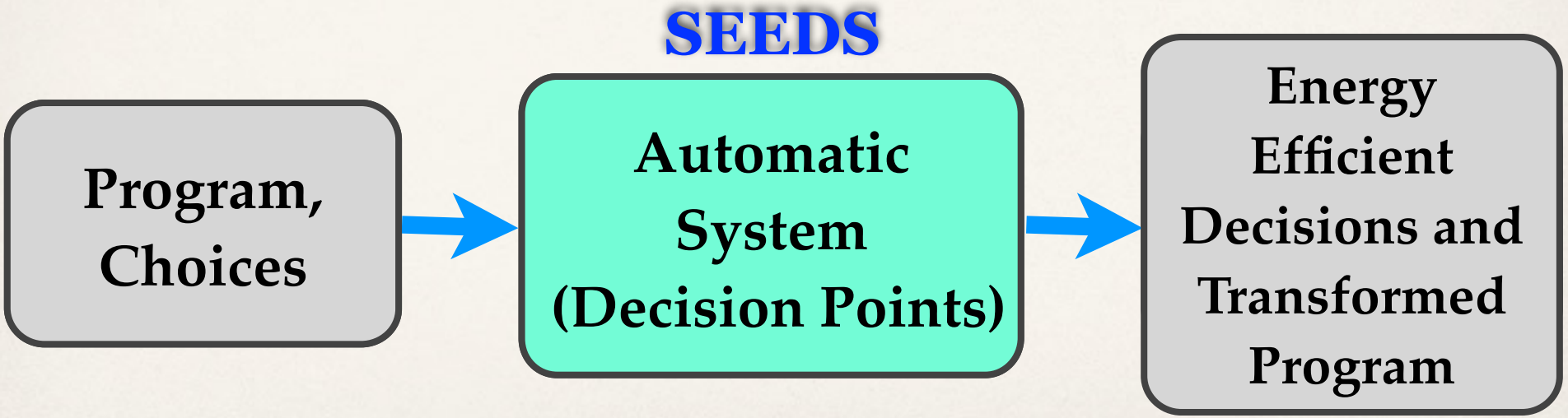
One decision is not enough!



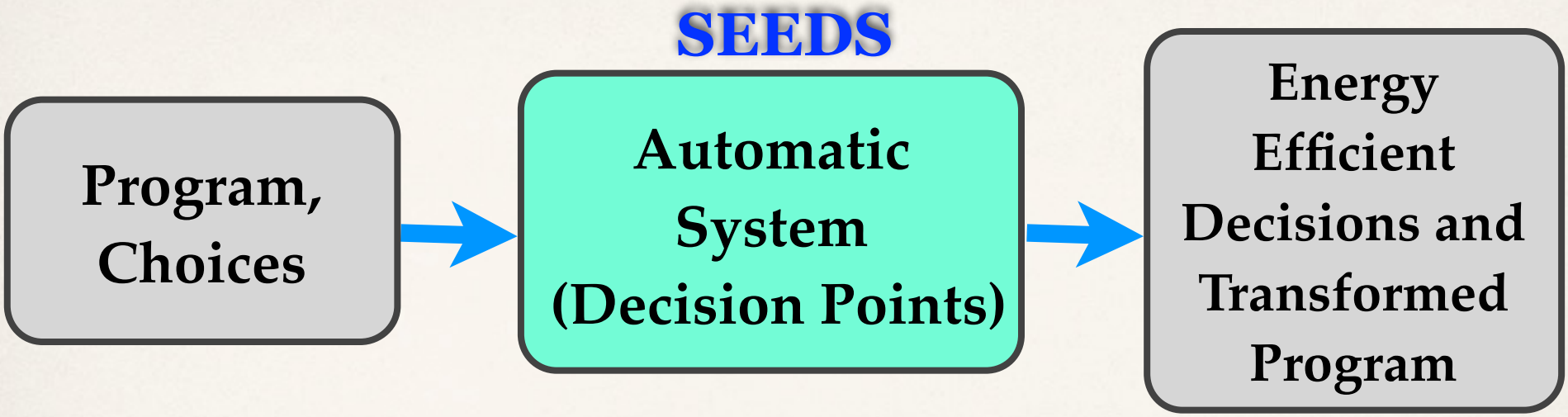
Who wants to make this many decisions?

How can one make the best selection in terms of energy efficiency ?

Targeted Research Question:
Is it possible to create a framework to automatically make these decisions for developers, or to support them in these decisions?



Research Challenges



(1) **Abstract away system/platform details/interactions** from developer decision making.

(2) **Automate identification** of potential choice space.

(3) **Automate monitoring of energy** usage and relation to decision choices.

(4) **Automate selection** of most energy efficient choices.

(5) **Support variety of software developer decisions**.

SEEDS Approach

(1) **Abstract away system/platform details/interactions** from developer decision making.

- ***Provide a framework to automatically identify energy efficient decisions.***

(2) **Automate identification** of potential choice space.

- ***Analyze the set of possible choices provided as input.***

(3) **Automate monitoring of energy** usage and relation to decision choices.

- ***Use hardware instrumentation technique to profile energy usage of choices.***

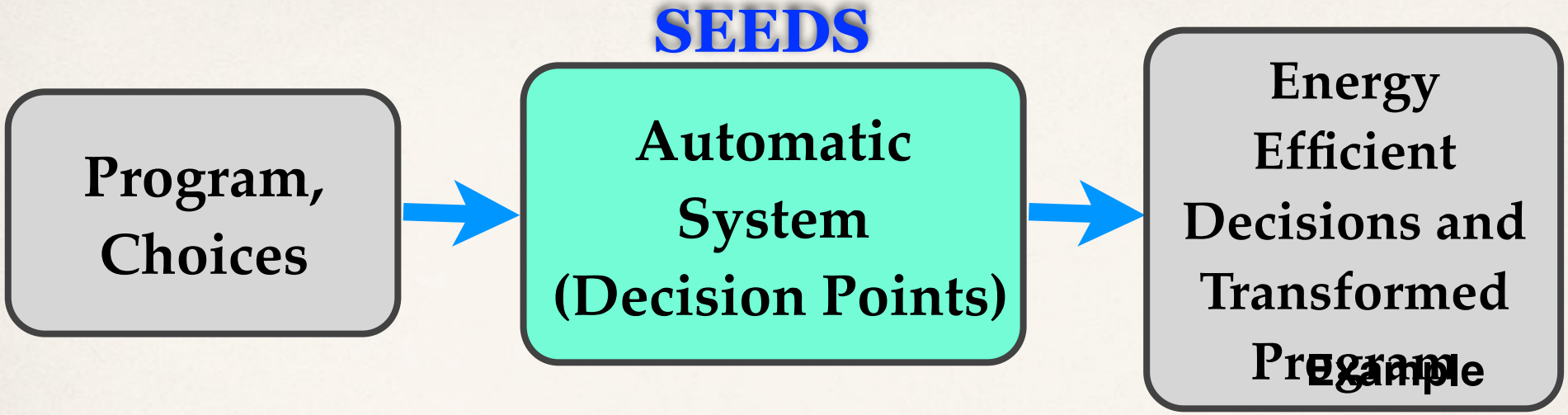
(4) **Automate selection** of most energy efficient choices.

- ***Analyze energy usage implications of choices.***

(5) **Support variety of software developer decisions**.

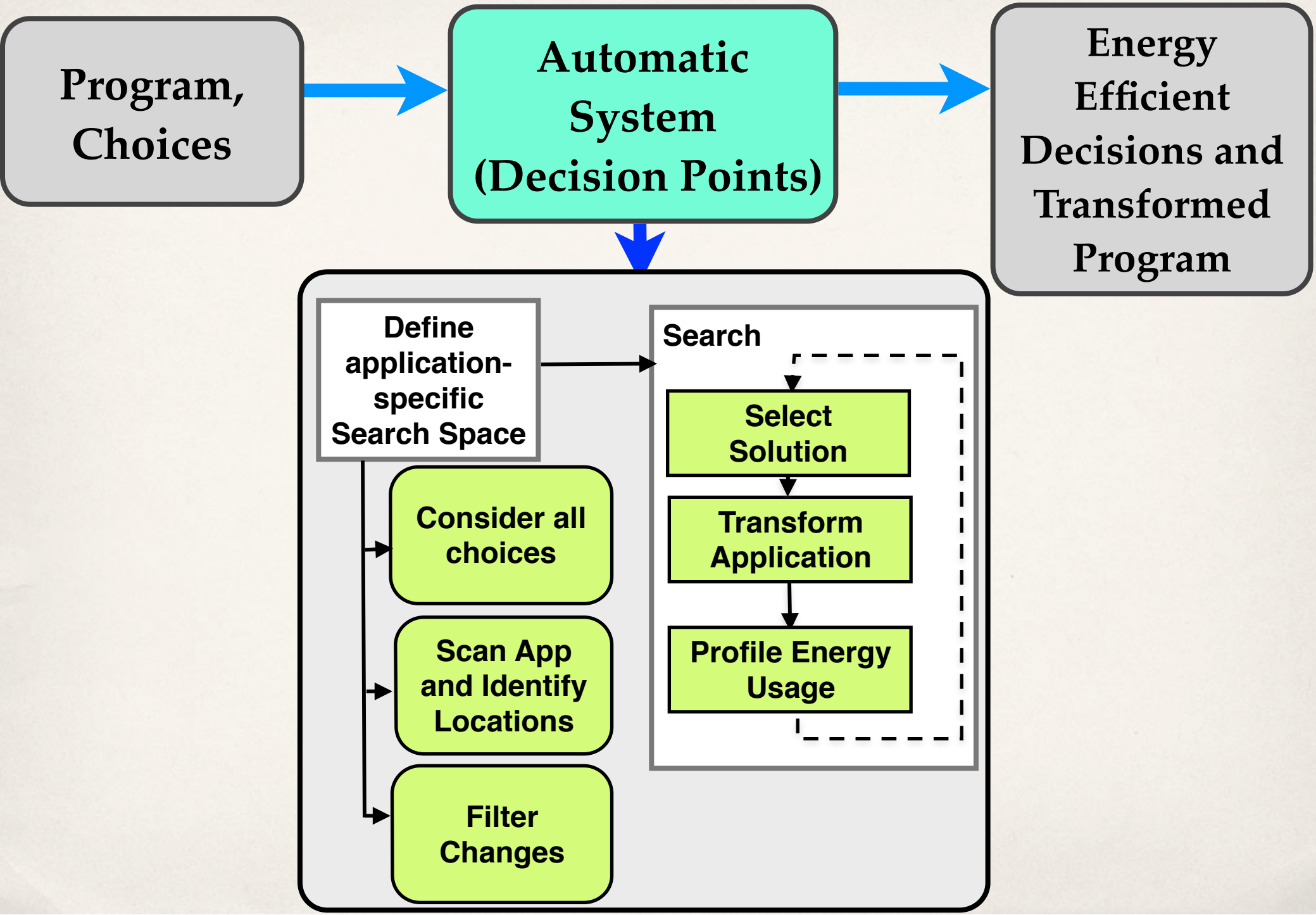
- ***Generalize inputs.***

Generalizing the Inputs

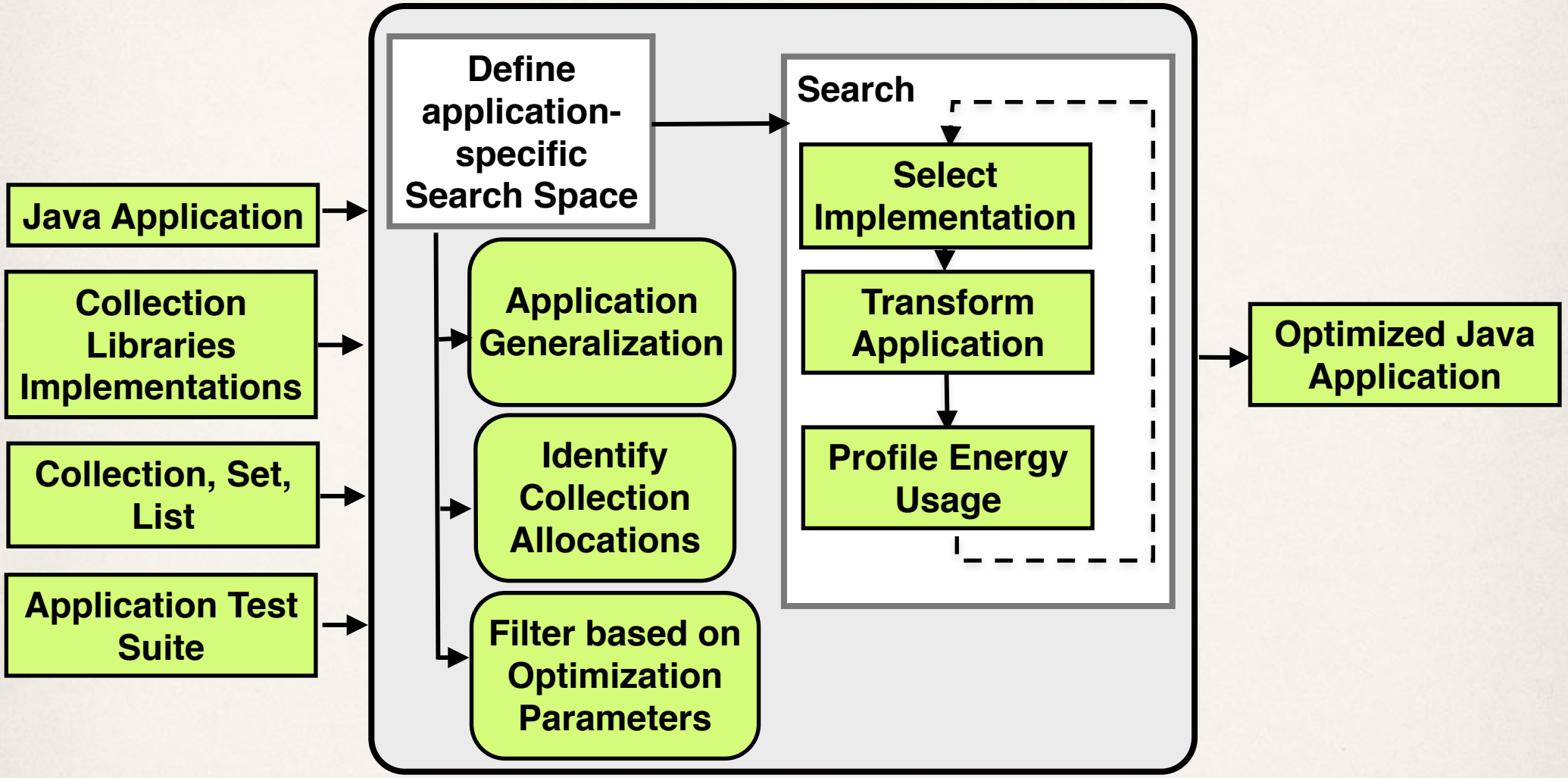


Application Code	SW application to be optimized for energy usage.	Java Application
Potential Changes	List of possible changes for the application.	Collection Libraries Implementations
Optimization Parameters	Constraints on potential changes.	Set, List
Context Information	Platform, expected inputs, workload.	Application Test Suite

SEEDS Framework Components



An Instantiation of SEEDS: SEEDS_api



SEEDS_api supports the selection of Library implementations to optimize the energy usage of a Java application.

Evaluation of SEEDS: Questions

- **Effectiveness**

To improve the energy usage of applications.

- **Exploration Capability**

To analyze and identify the best choices.

- **Cost**

Time required to find the best solution.

Methodology for Evaluation

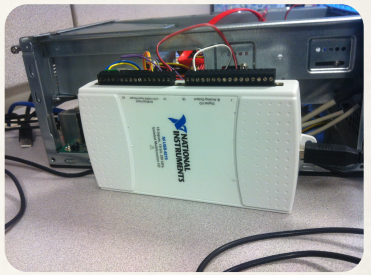
- **Independent Variable:**

  Library Implementations

- **Dependent Variable:**

  Energy usage [Joules]

- **Subjects:**



7 Java Applications

Barbecue, Jdepend, Apache-XML-Security, Jodatime, Commons-Lang, Commons-beanutils, Commons-cli

Software-artifact Infrastructure Repository

source forge

Apache Commons

6 Collections Libraries

:::fastutil:

javolution

commons collections

Goldman Sachs

JCF

Impact of Search Space Size

Effectiveness of SEED

% Improvement

Application	Optimization with JCF Only	Optimization with All Libraries	
Barbecue	17*	17*	✓
Jdepend	3*	6*	✓ 3 %
Apache-xml-security	5	5	✓
JodaTime	8*	9	✓ 1 %
Commons-lang	10	13	✓ 3 %
Commons-beanutils	-	-	✗
Commons-cli	2*	2*	✓

* only one change † multiple changes

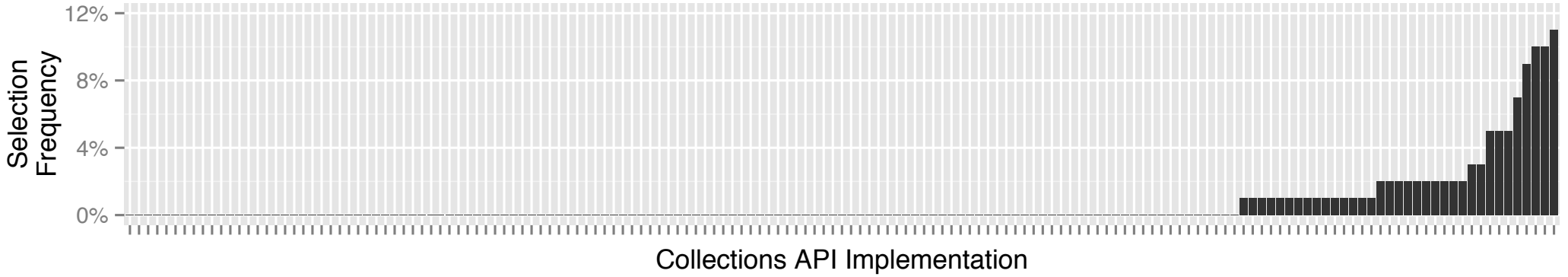
Sometimes having more choices to select the best implementation results in an improvement.

Using SEEDS: How often developers choose energy efficient implementations?

- SEEDS used to analyze **123** targeted locations for a possible change of implementation.
- For JCF Implementations, **56%** (69 cases) when switching from original implementation **improved energy usage**.
- All Libraries, **72%** (89 cases) when switching from original **improved the energy usage**

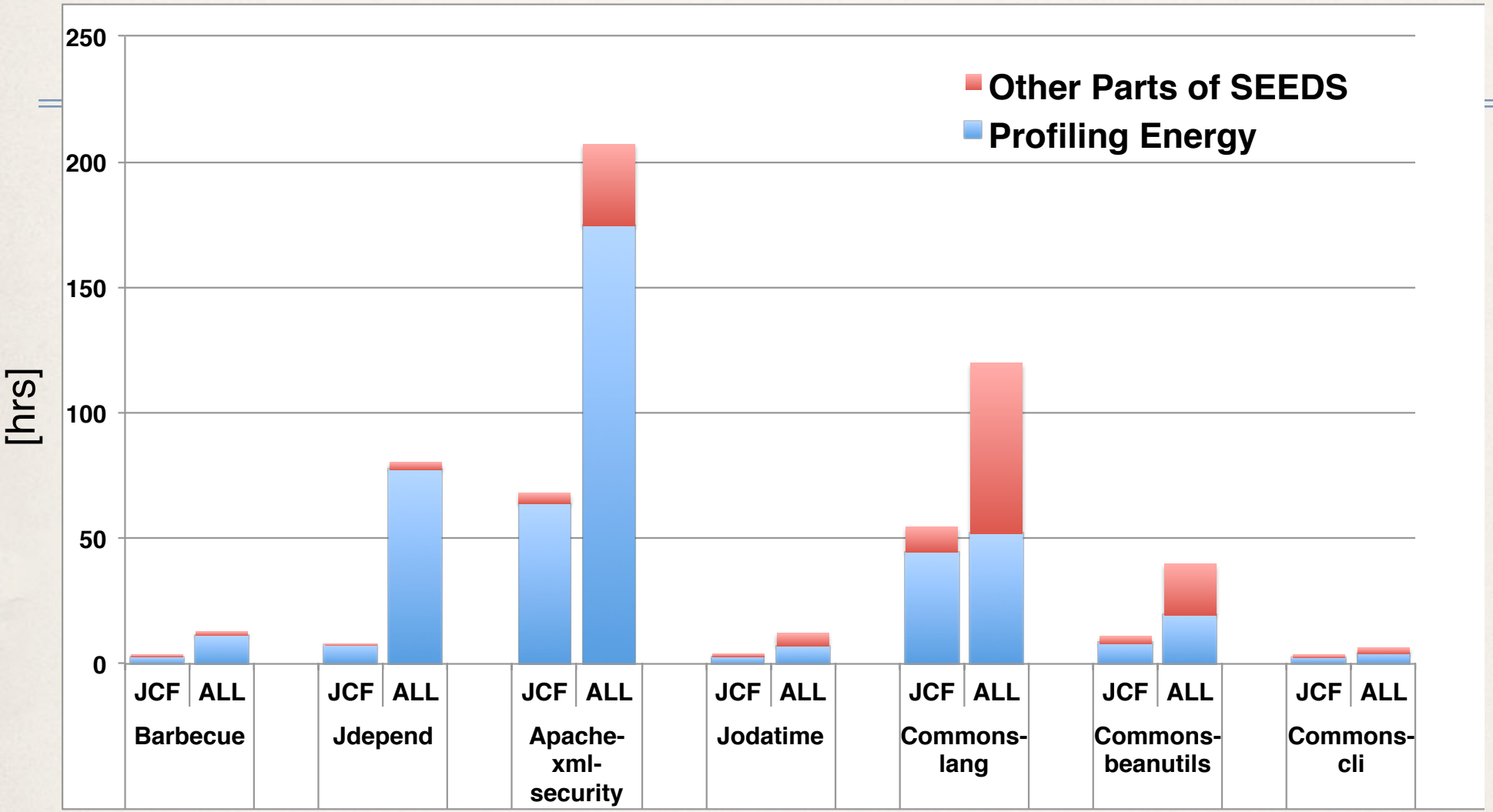
How often is a particular implementation the most energy efficient?

157 distinct implementations were selected as energy efficient options from Collections API, from All libraries.



Implementations that were most frequently the most energy efficient: ArrayList and HashSet.

Cost of SEEDS for each application



Most costly part is profiling the energy usage.

Related Work

- **Empirical Case Studies**

  Insights about 'possible' impacts of some algorithms, design patterns, etc.(e.g., [C. Bunse, 2009][C. Sahin, 2012]*).*

  Not applicable across contexts.

- **Measurements of Energy Consumption of Applications**

  Provide energy usage of application by different approaches (HW instrumentation, simulation, estimation) (e.g., [D. Singh, 2010][S. Hao, 2013]*.*

  Do not provide information about which changes could be made to reduce the energy.

- **Language Support**

  Implement energy management policies according to device status or tasks being executed (e.g., [M. Cohen, 2012]*).*

  Need to Know which policies to implement, where and when to apply them manually.

Conclusions and Future Work

- It is hard to establish and manually modify high level code that takes into account energy usage implications.
- SEEDS supports developers to automatically make changes that produce more energy efficient applications.
- SEEDS can effectively find the energy efficient version of an application with respect to potential changes.
- Future work will be focused in improve SEEDS and its search strategy.

Thanks!

Questions?

imanotas@udel.edu