Data-Oriented Characterization of Application-Level Energy Optimization

Kenan Liu[1], Gustavo Pinto[2], and Yu David Liu[1]

[1]State University of New York, Binghamton
[2] Federal University of Pernambuco
FASE, 2015

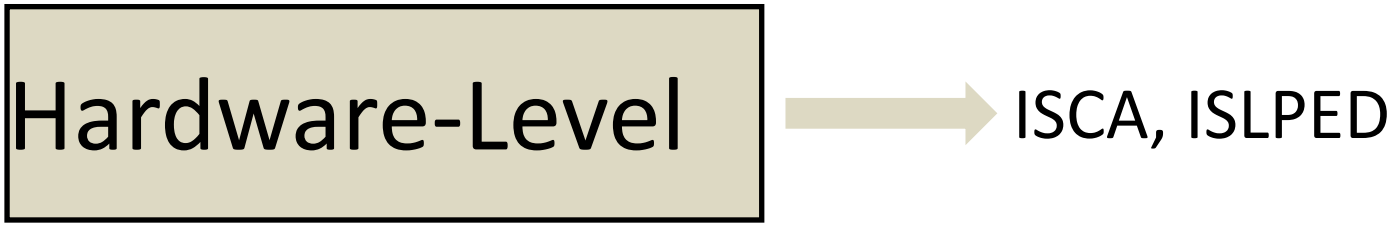
BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK

Why?

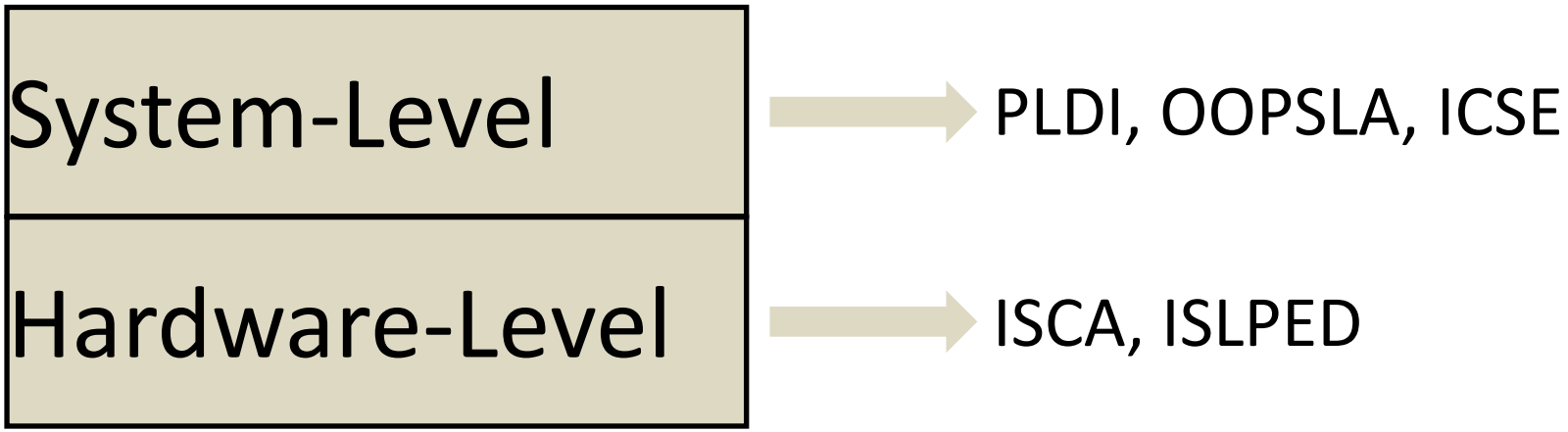
- Mobile devices

- Data center

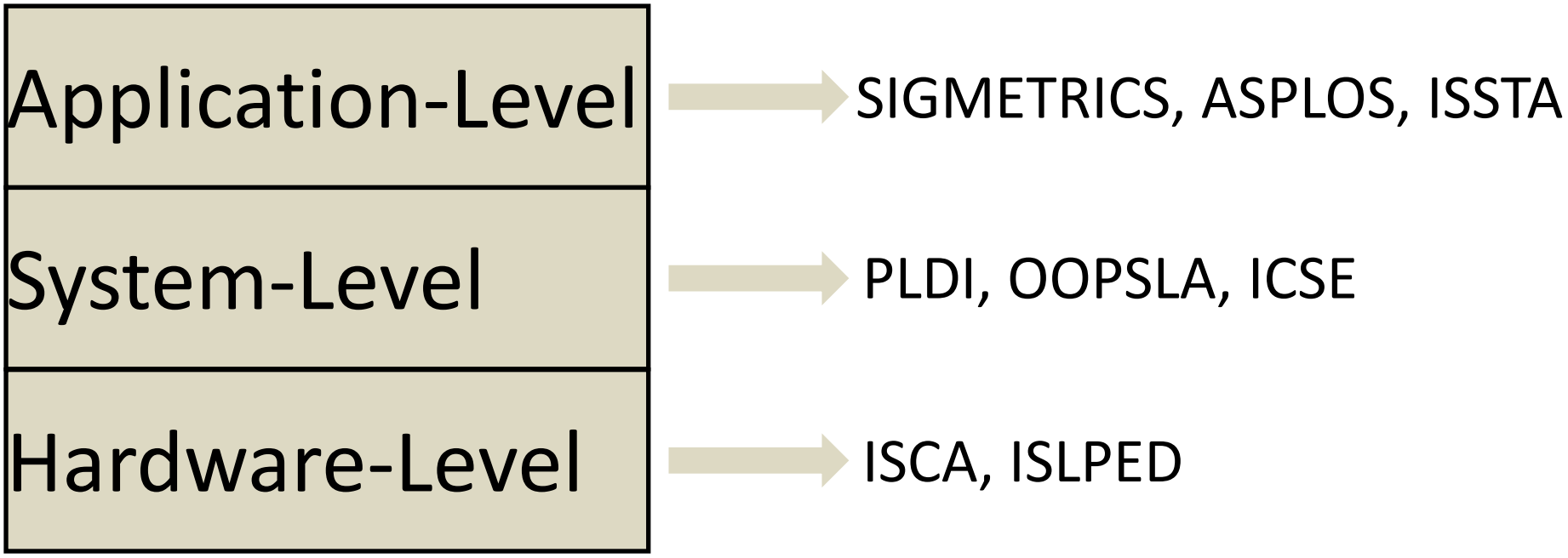
State of the Art



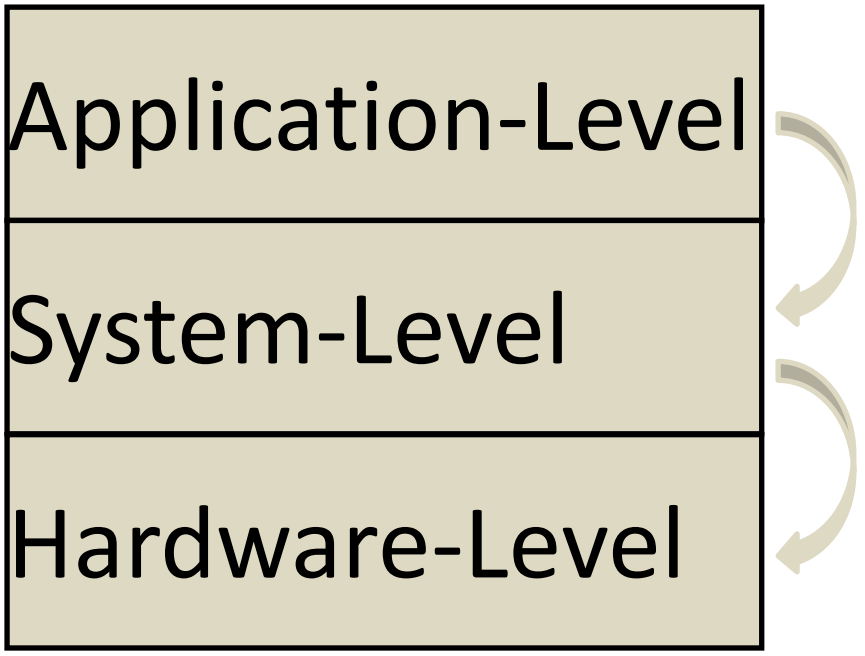
State of the Art

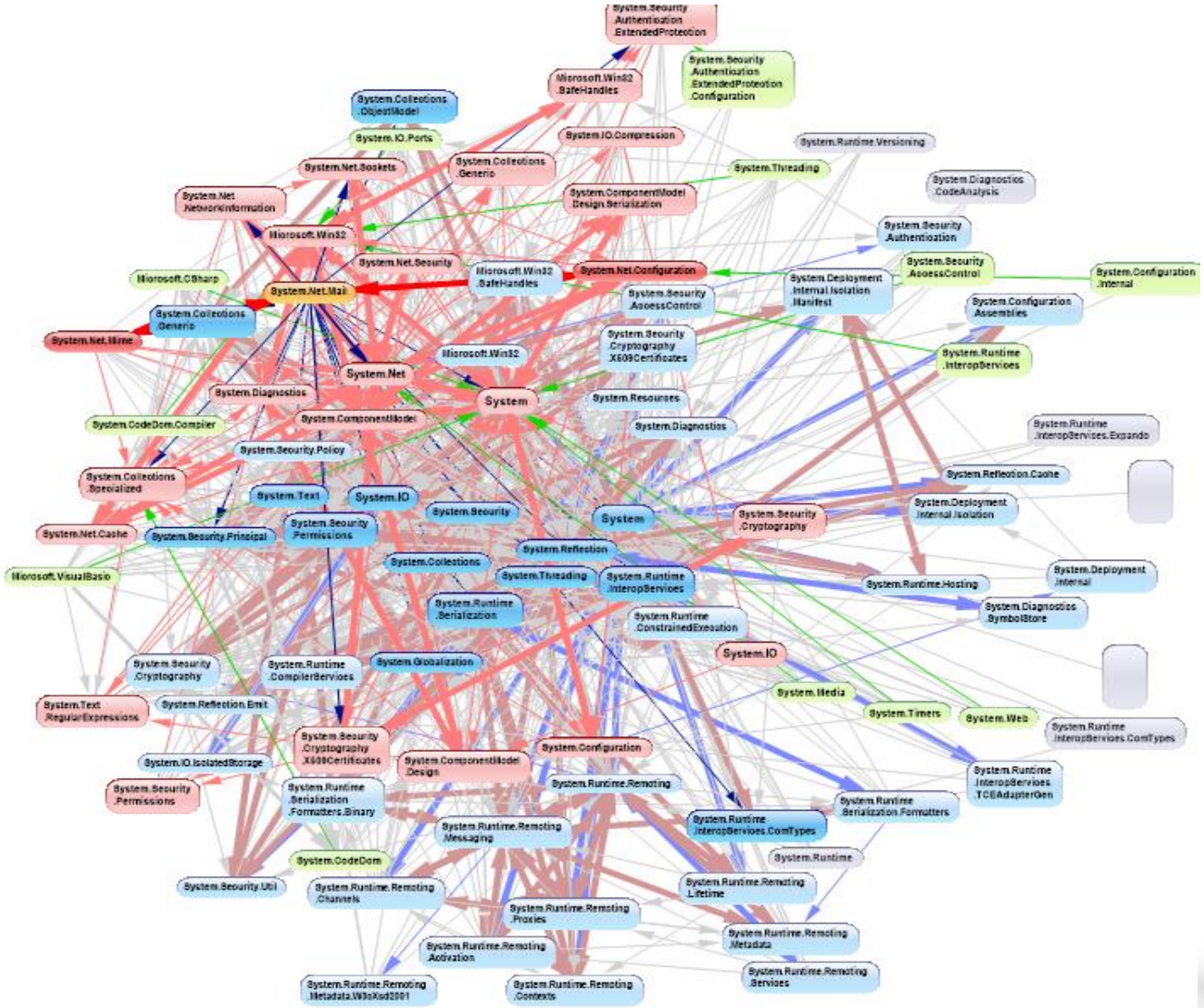


State of the Art

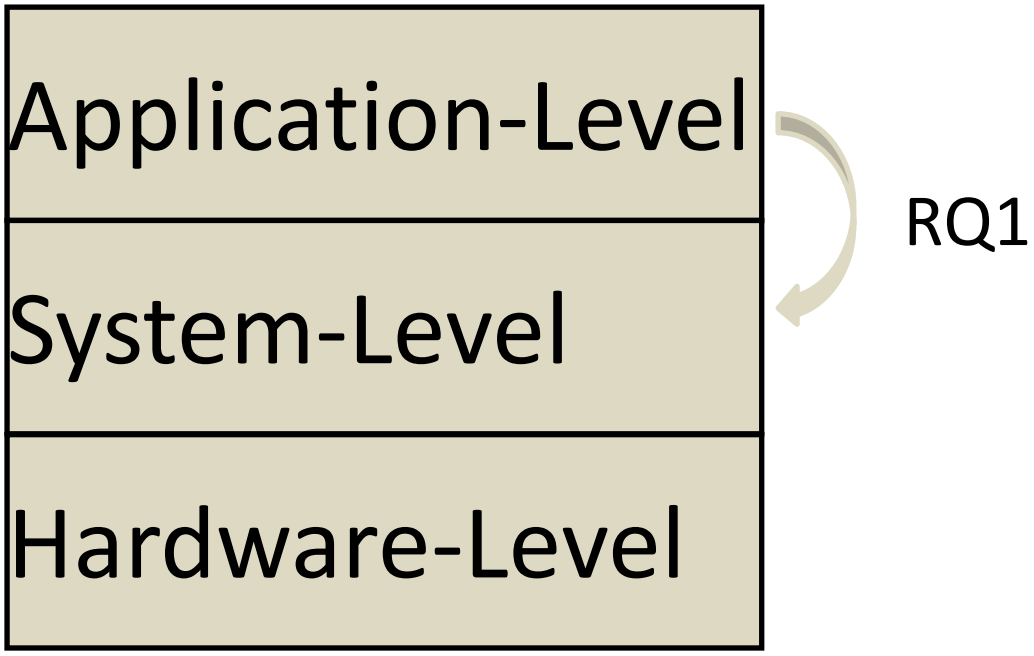


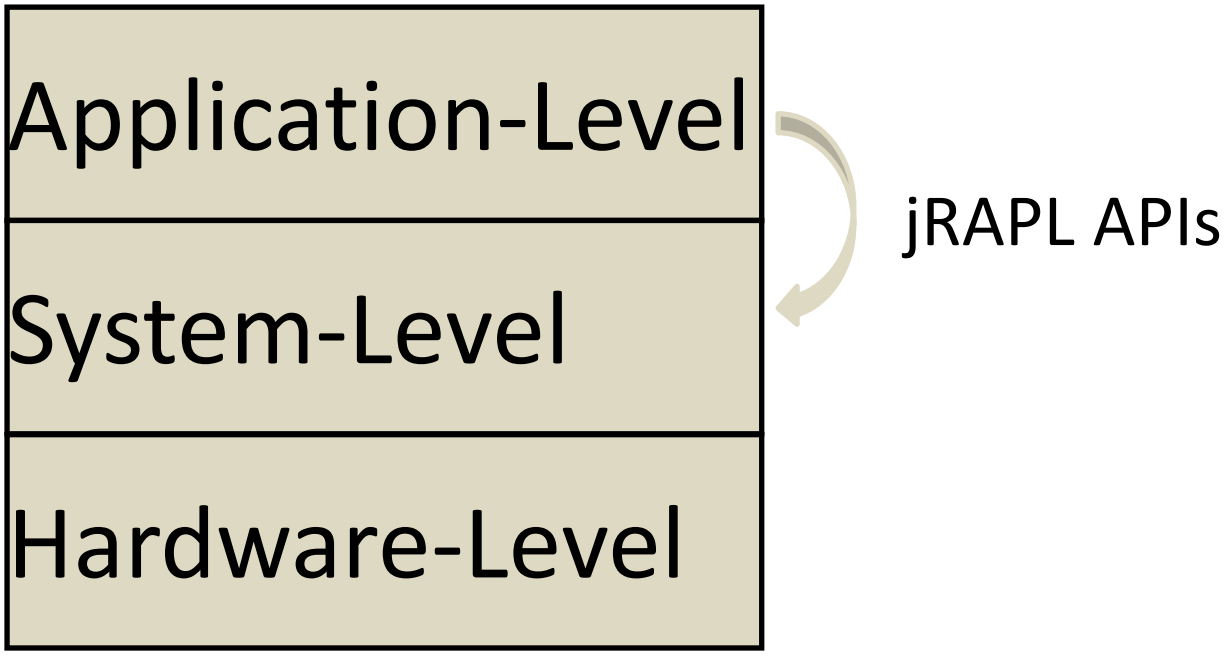
State of the Art

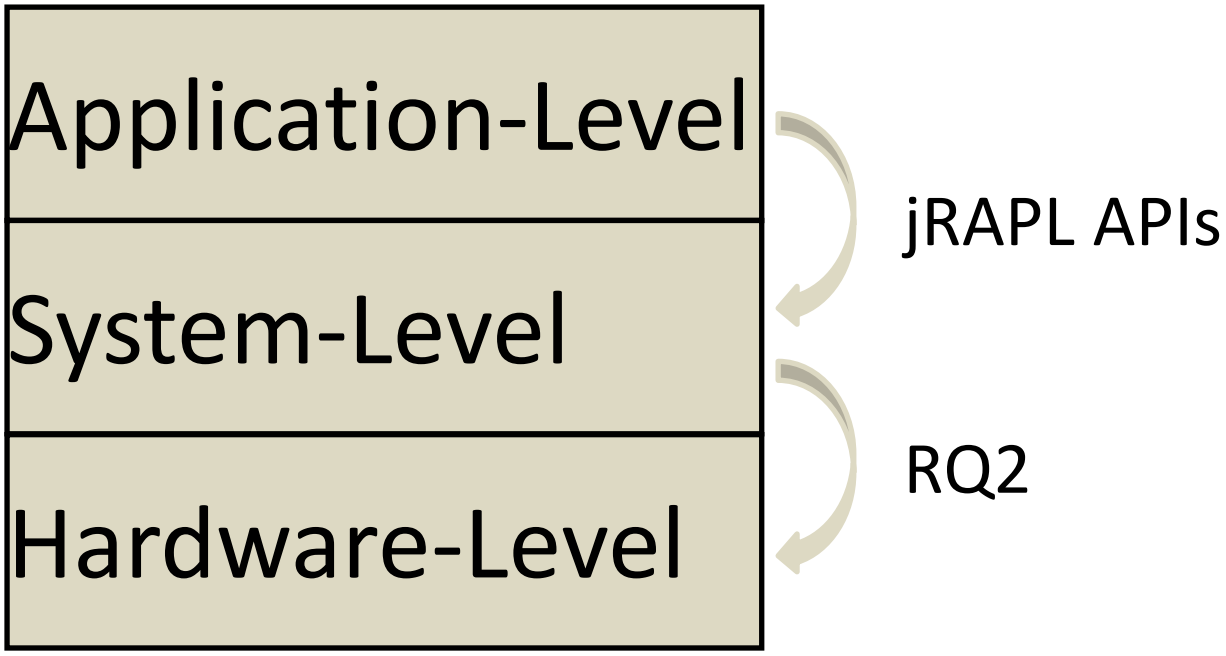


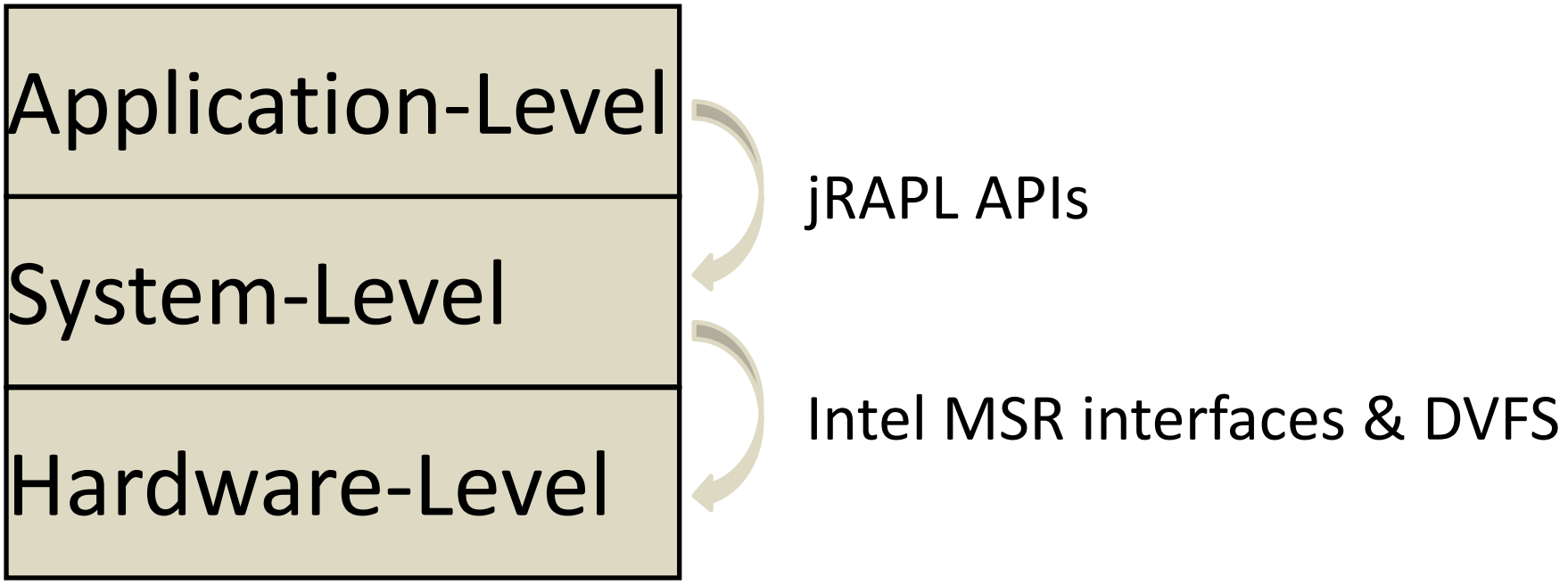

System.Security.Authentication.ExtendedProtection
Microsoft.Win32.SafeHandles
System.Security.Authentication.ExtendedProtection.Configuration
System.Collections.ObjectModel
System.IO.Ports
System.IO.Compression
System.Runtime.Versioning
System.Net.Sockets
System.Collections.Generic
System.Threading
System.Diagnostics.CodeAnalysis
System.Net.NetworkInformation
System.ComponentModel.Design.Serialization
Microsoft.Win32
System.Security.Authentication
Microsoft.CSharp
System.Net.Security
Microsoft.Win32.SafeHandles
System.Net.Configuration
System.Security.AccessControl
System.Configuration.Internal
System.Net.Mail
System.Deployment.Internal.Isolation.Manifest
System.Security.AccessControl
System.Collections.Generic
System.Configuration.Assemblies
System.Net.Mime
System.Security.Cryptography.X509Certificates
Microsoft.Win32
System.Runtime.InteropServices
System.Net
System.Diagnostics
System
System.Resources
System.CodeDom.Compiler
System.ComponentModel
System.Diagnostics
System.Runtime.InteropServices.Expando
System.Security.Policy
System.Collections.Specialized
System.Reflection.Cache
System.Text
System.IO
System.Security
System
System.Security.Cryptography
System.Deployment.Internal.Isolation
System.Net.Cache
System.Security.Principal
System.Security.Permissions
System.Reflection
System.Collections
System.Threading
System.Runtime.InteropServices
System.Runtime.Hosting
System.Deployment.Internal
Microsoft.VisualBasic
System.Runtime.Serialization
System.Diagnostics.SymbolStore
System.Runtime.ConstrainedExecution
System.IO
System.Security.Cryptography
System.Runtime.CompilerServices
System.Globalization
System.Media
System.Text.RegularExpressions
System.Reflection.Emit
System.Timers
System.Web
System.Runtime.InteropServices.ComTypes
System.Security.Cryptography.X509Certificates
System.Configuration
System.IO.IsolatedStorage
System.ComponentModel.Design
System.Runtime.Remoting
System.Runtime.InteropServices.TCEAdapterGen
System.Security.Permissions
System.Runtime.Serialization.Formatters.Binary
System.Runtime.Serialization.Formatters
System.Runtime.Remoting.Messaging
System.Runtime.InteropServices.ComTypes
System.CodeDom
System.Runtime
System.Security.Util
System.Runtime.Remoting.Channels
System.Runtime.Remoting.Lifetime
System.Runtime.Remoting.Proxies
System.Runtime.Remoting.Metadata
System.Runtime.Remoting.Activation
System.Runtime.Remoting.Services
System.Runtime.Remoting.Metadata.W3cXsd2001
System.Runtime.Remoting.Contexts

- RQ1 How does the choice of application-level data management features impact energy consumption?
- RQ2 How does application-level energy management interact with hardware-level energy management?

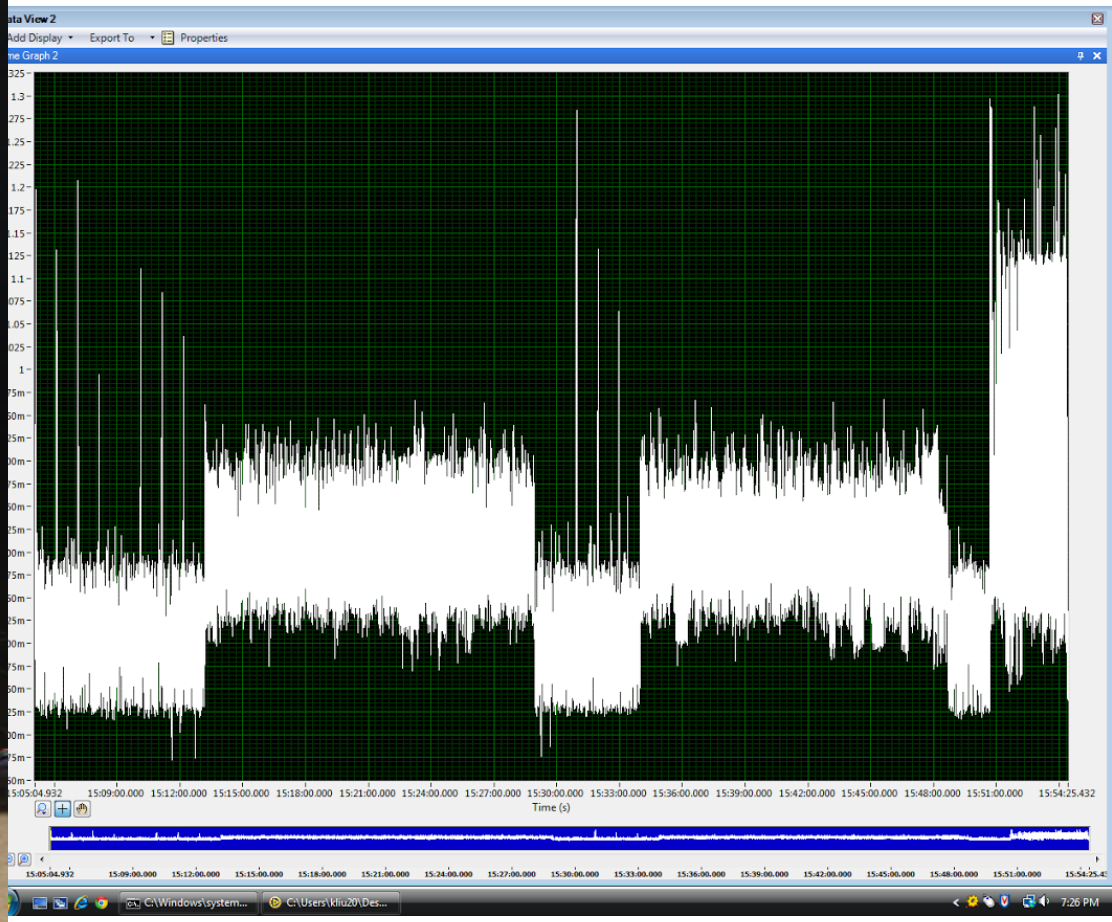

Application-Level
System-Level
Hardware-Level
RQ1


Application-Level
System-Level
Hardware-Level
jRAPL APIs

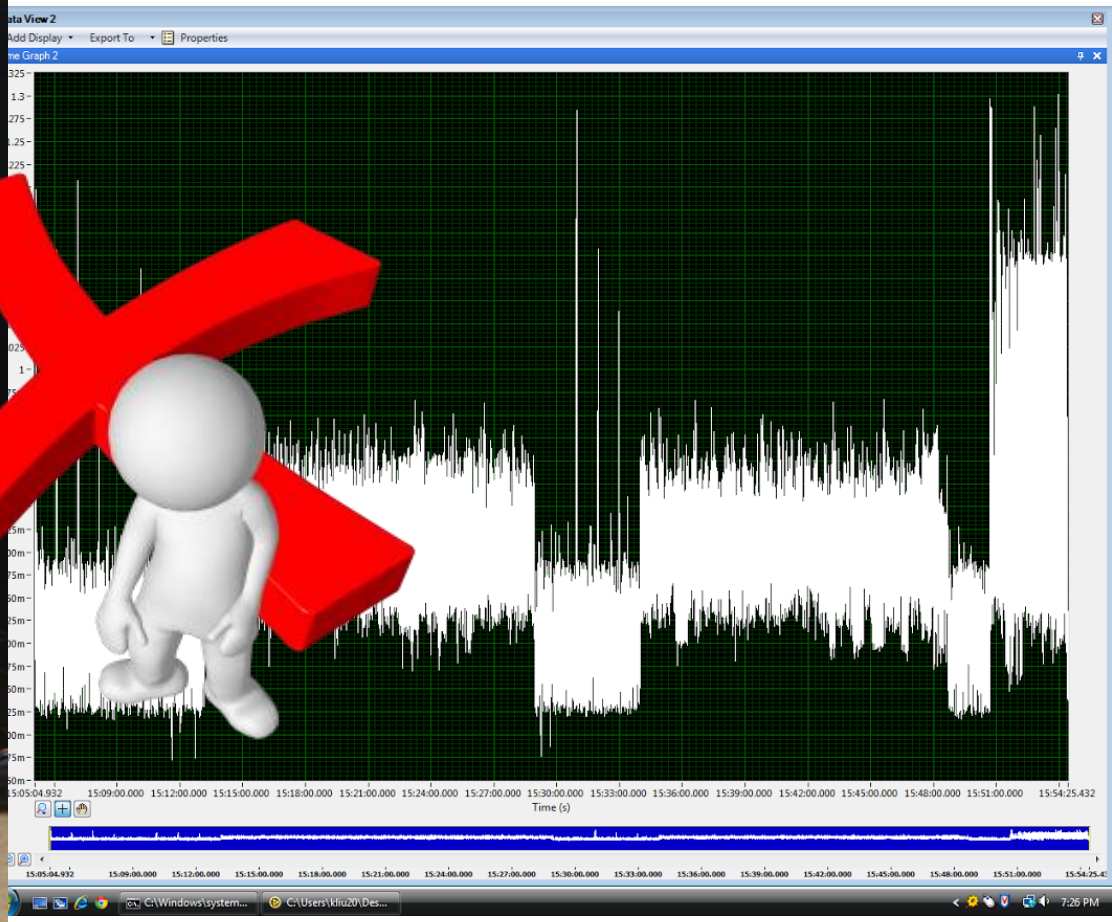

Application-Level
System-Level
Hardware-Level
jRAPL APIs
RQ2


Application-Level
System-Level
Hardware-Level
jRAPL APIs
Intel MSR interfaces & DVFS

Methodology



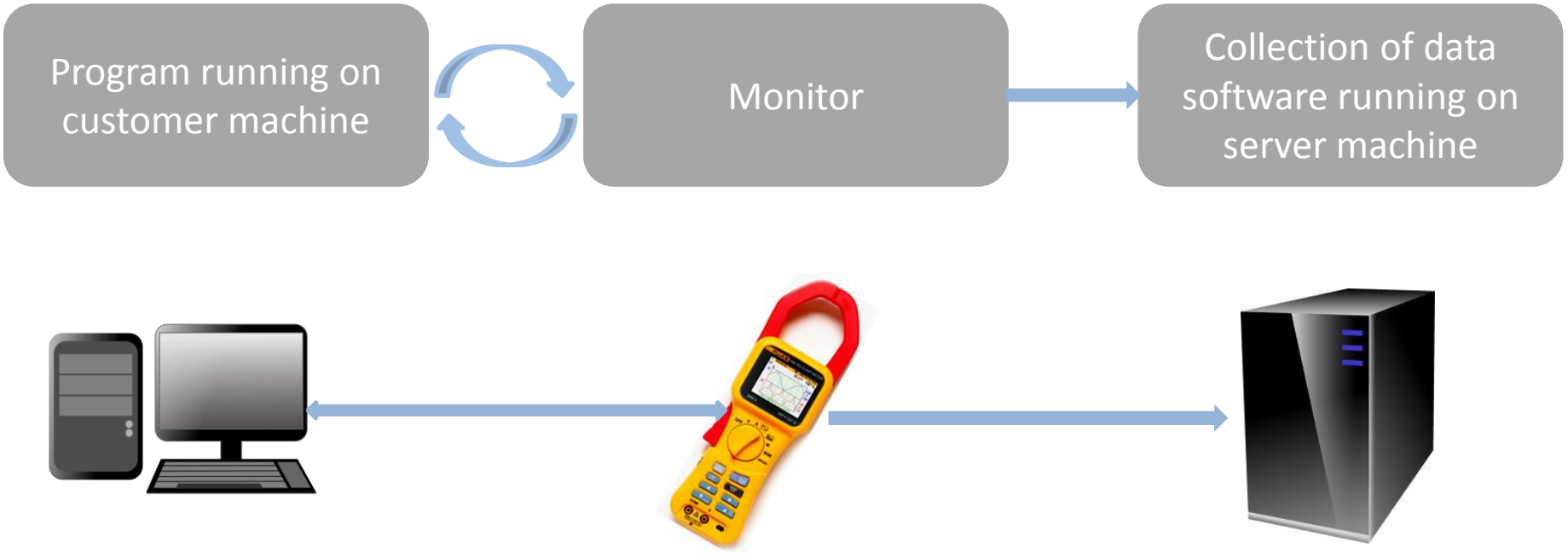
Methodology



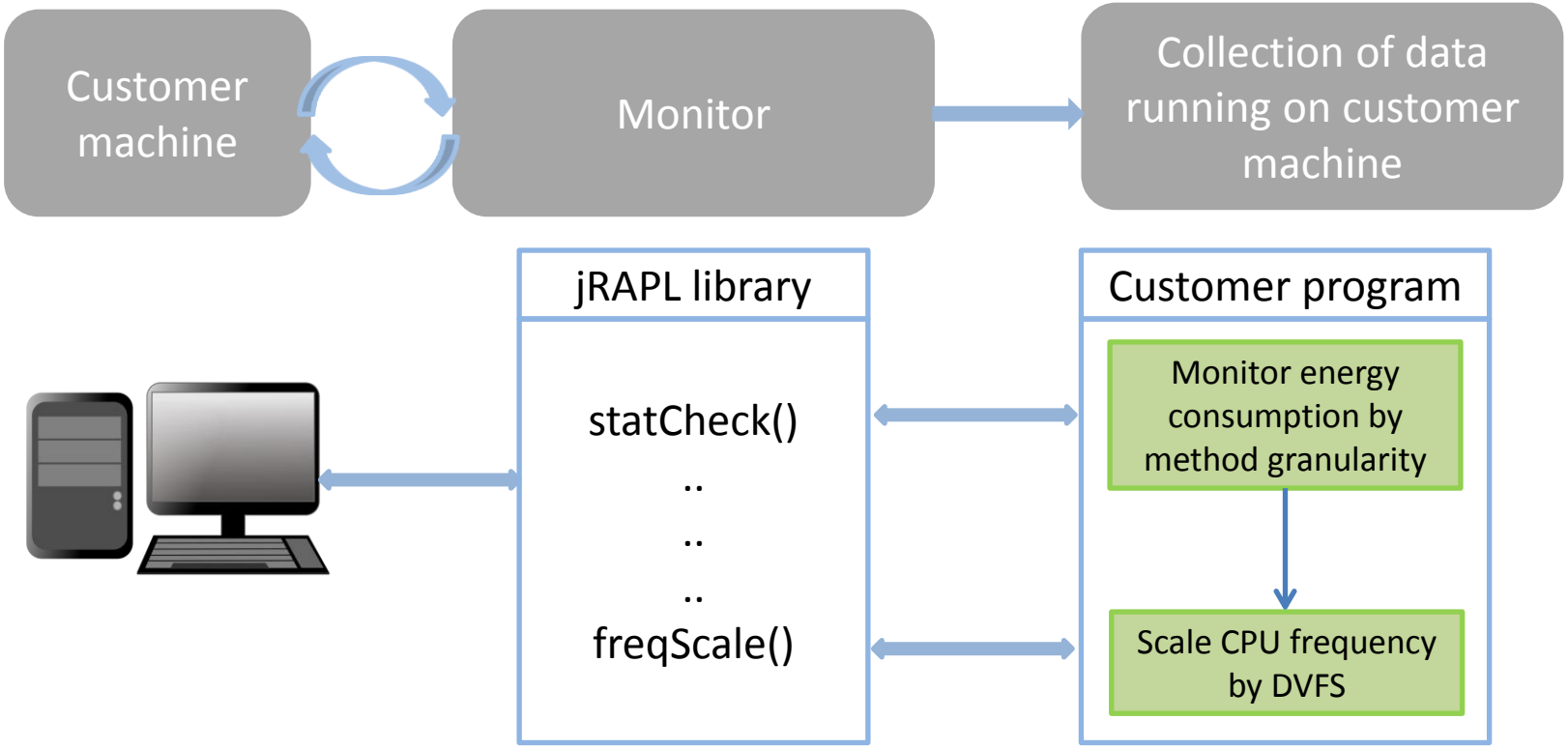
Methodology

```
double beginning = EnergyCheck.statCheck();
doWork();
double end = EnergyCheck.statCheck();
```

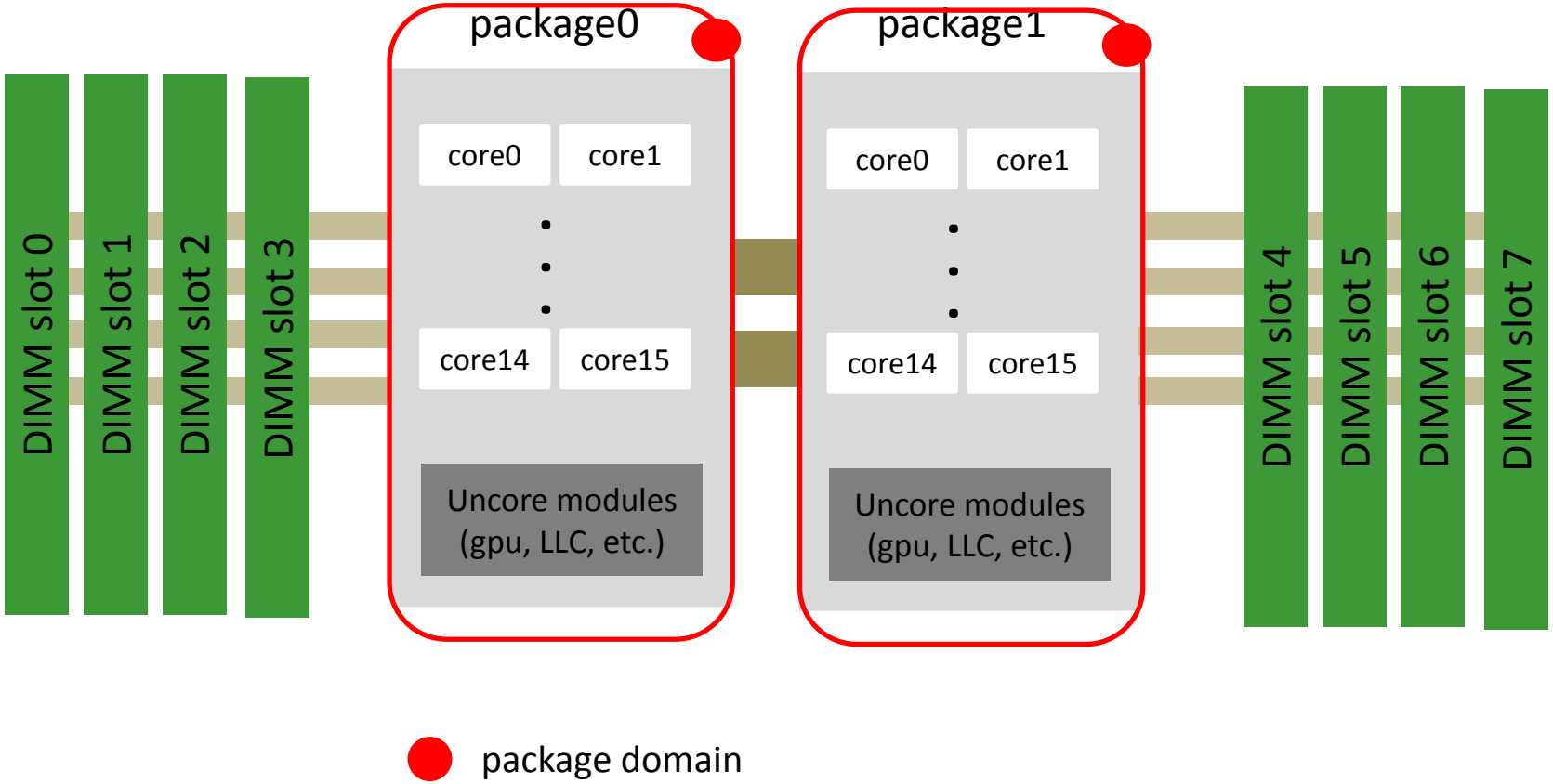
Traditional methodology – meter-based



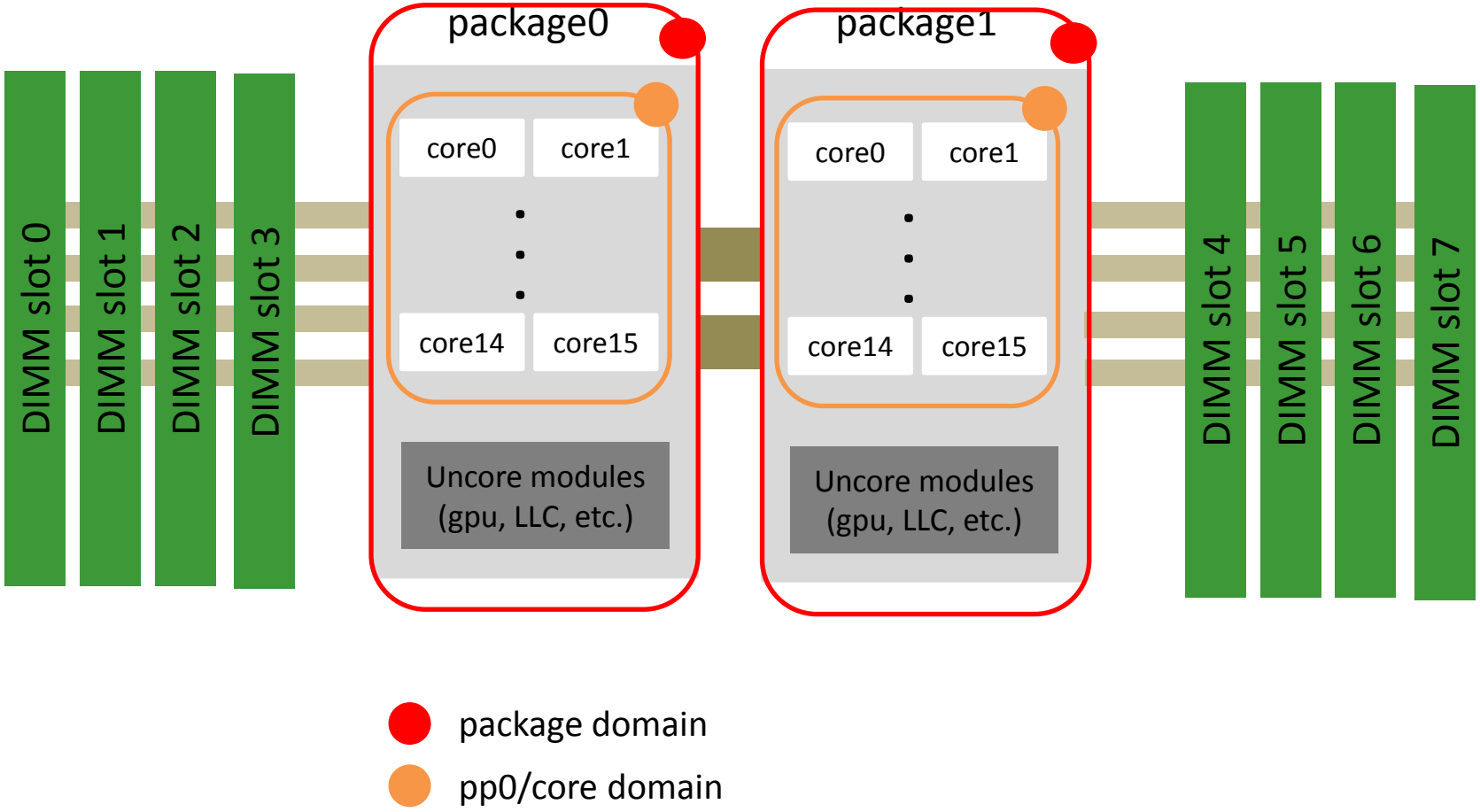
Methodology – jRAPL



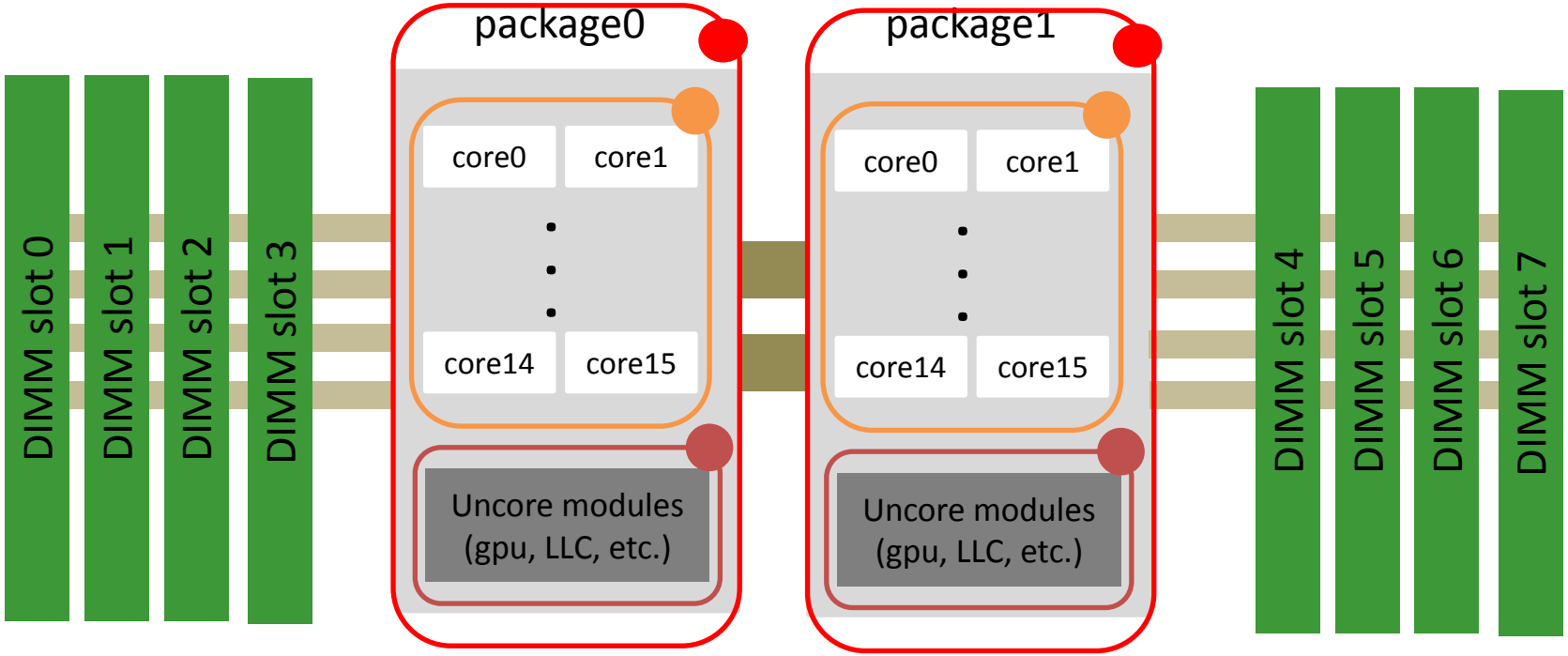
Methodology – jRAPL



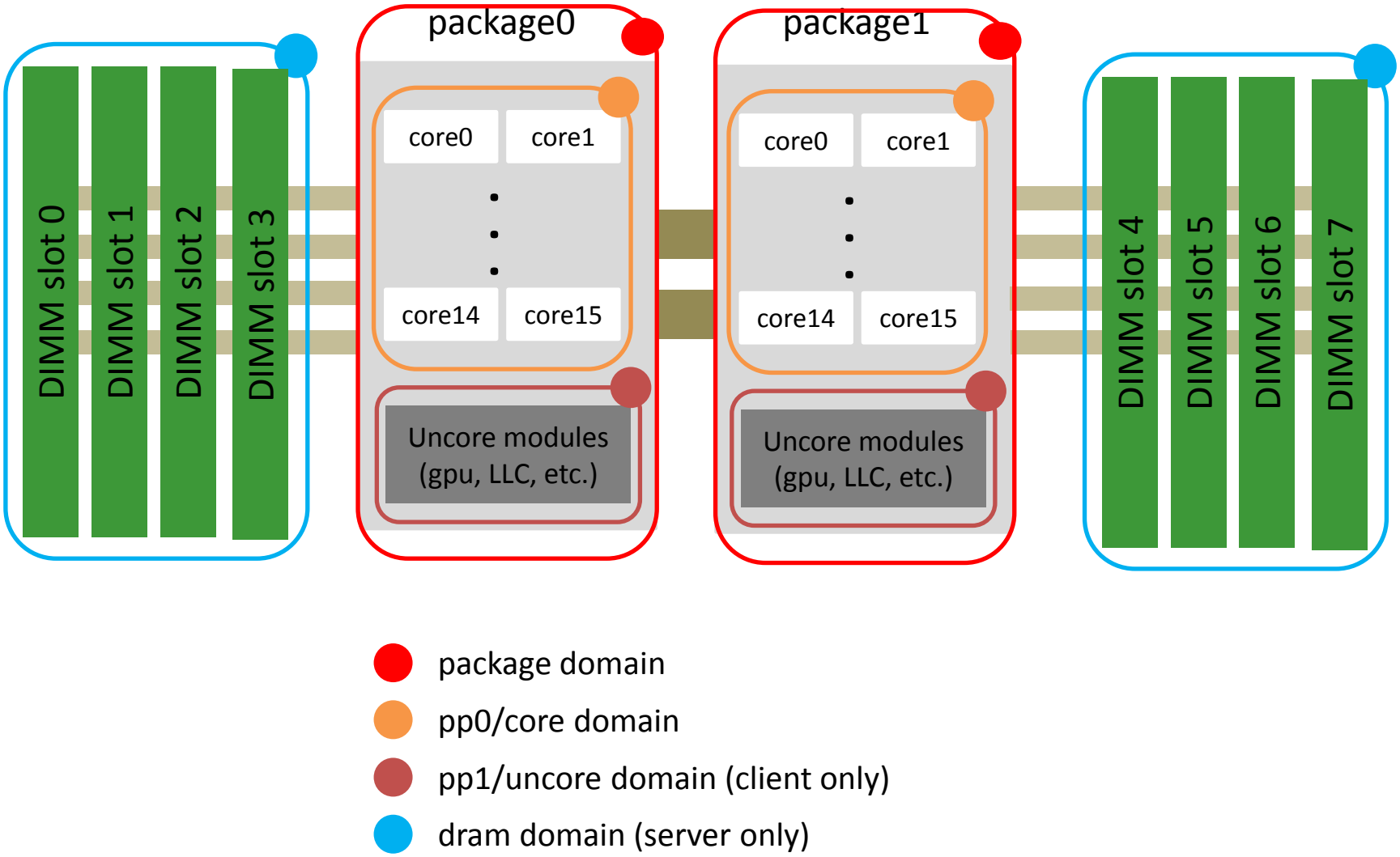
Methodology – jRAPL



Methodology – jRAPL



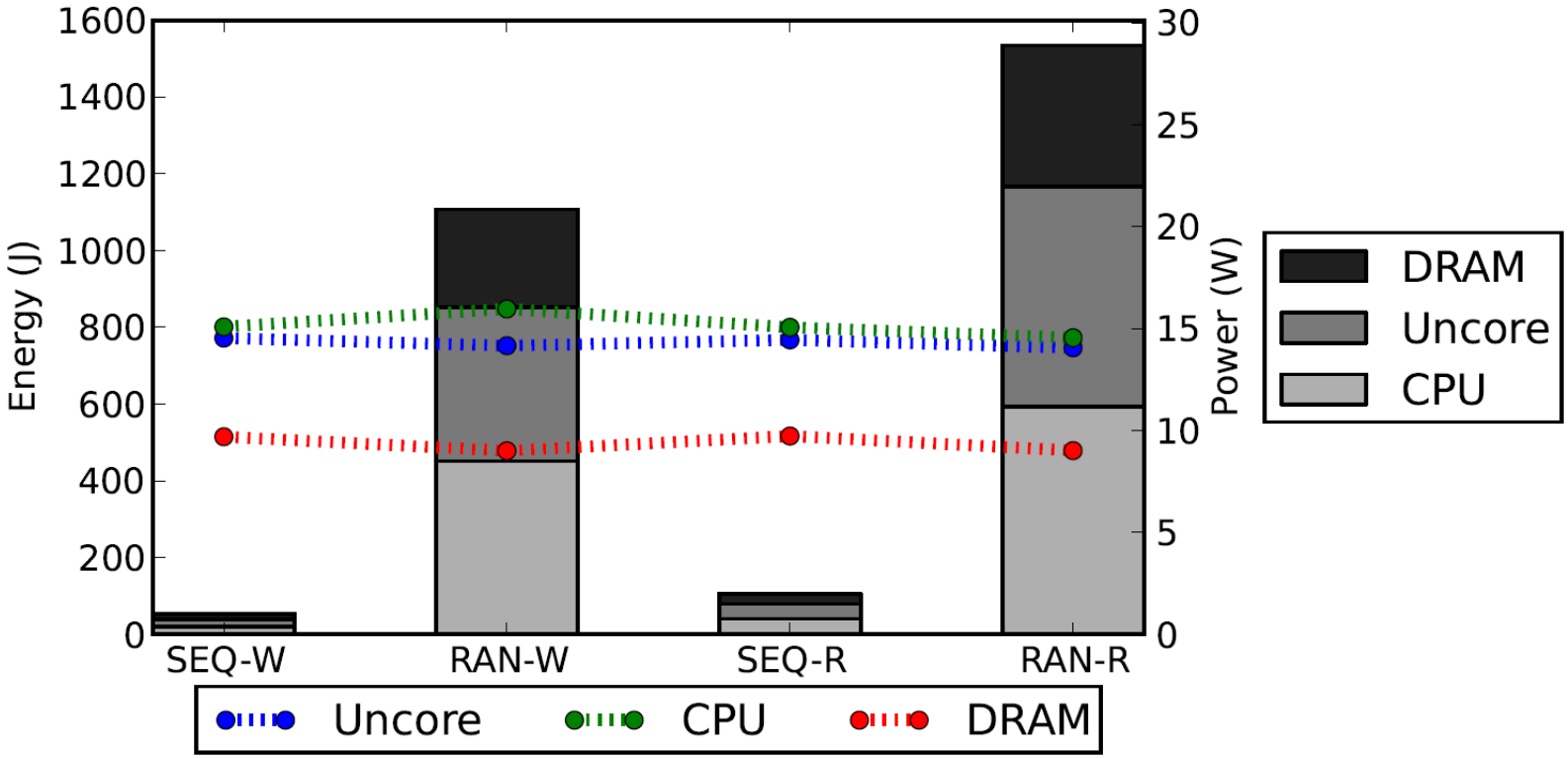
Methodology – jRAPL



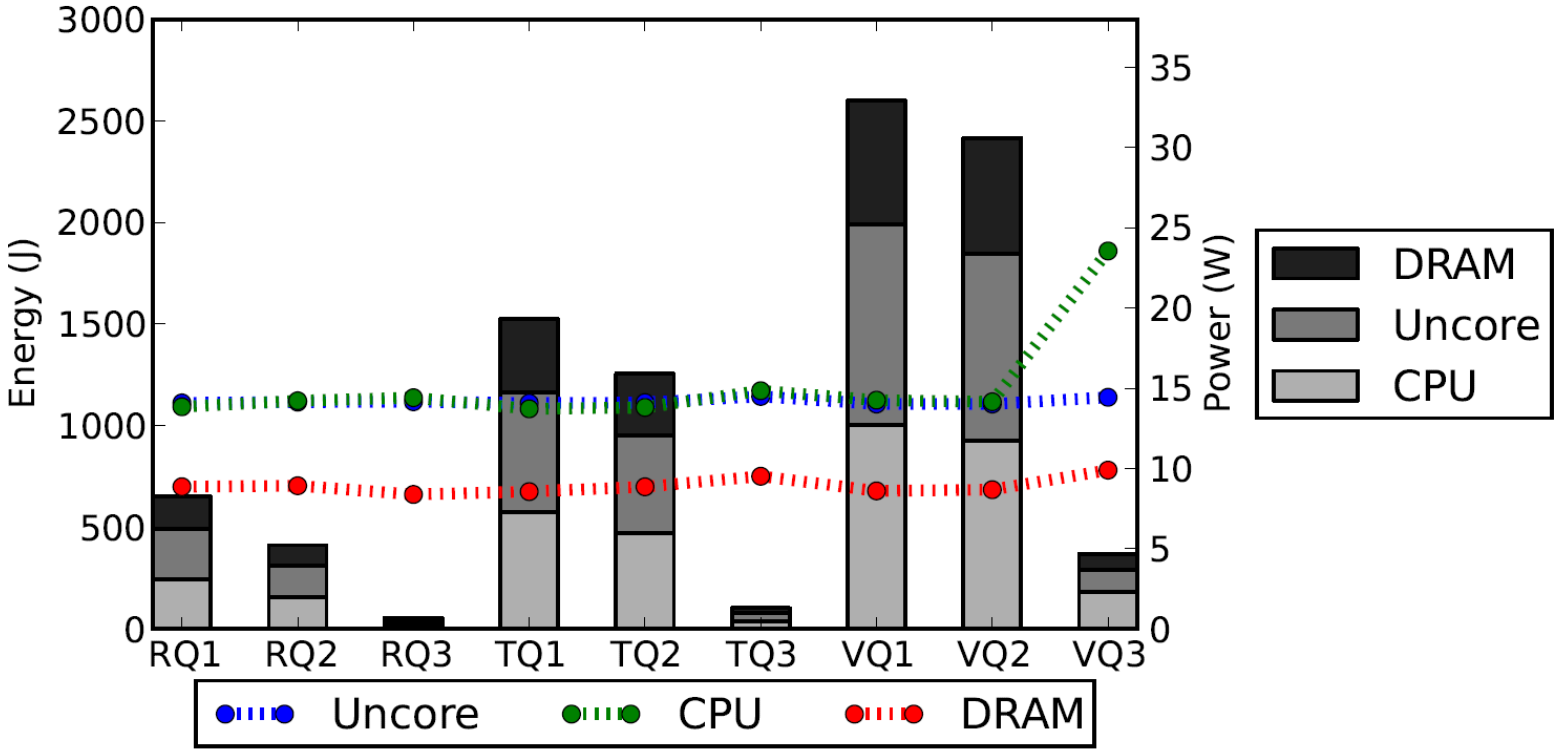
RQ1 Programmer Choices

- **Data access pattern**: sequential vs. random, read vs. write
- **Data organization and representation**: reference vs. data vs. type query, unboxed vs. boxed data, a primitive array vs. an ArrayList, object-centric vs. attribute-centric data grouping
- **Data precision**: short, int, float, double and long
- **Data I/O strategies**: buffering vs. unbuffering

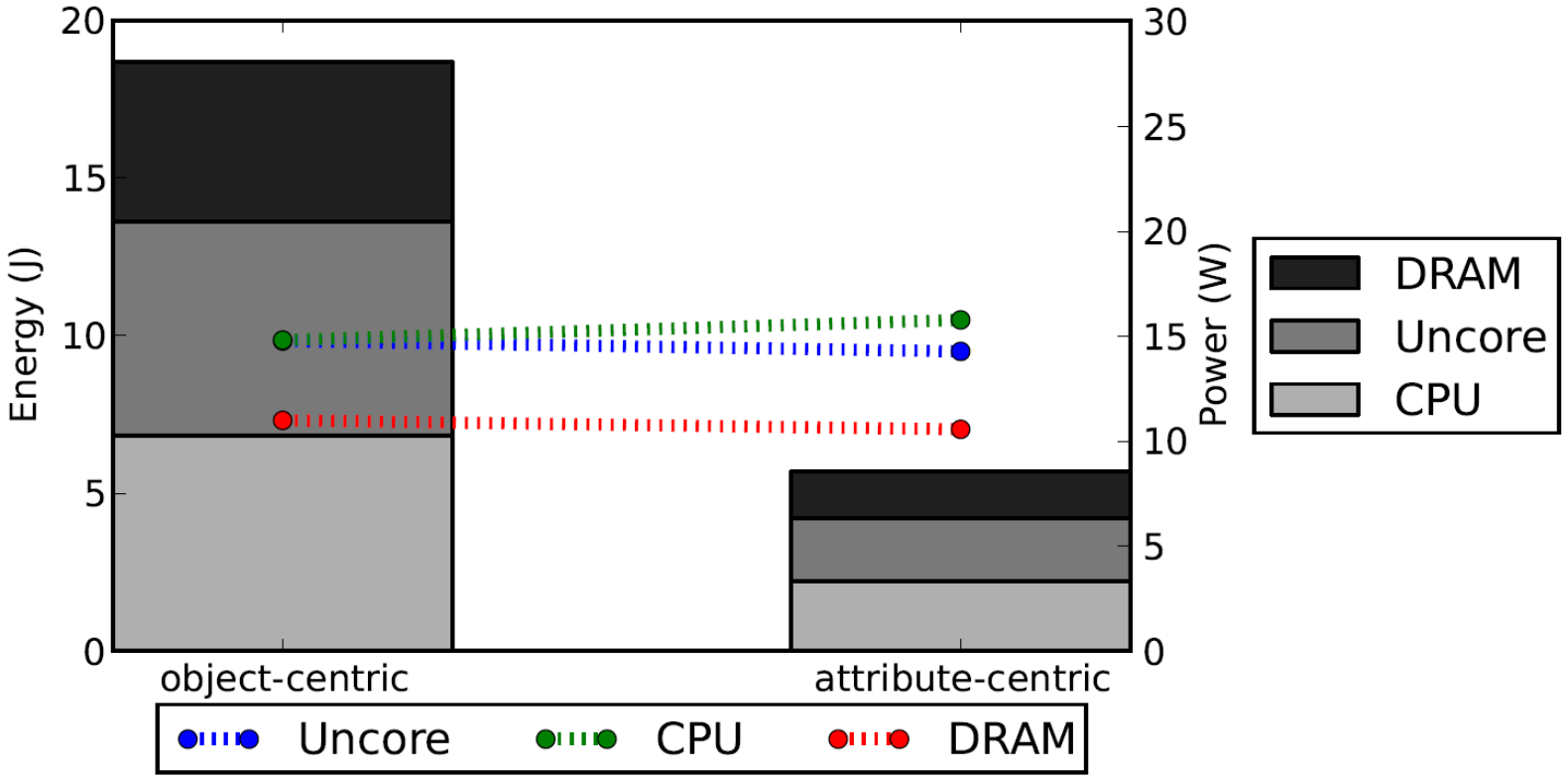
Data Representation Strategies



Data Representation Strategies



Data Organization



Data Organization

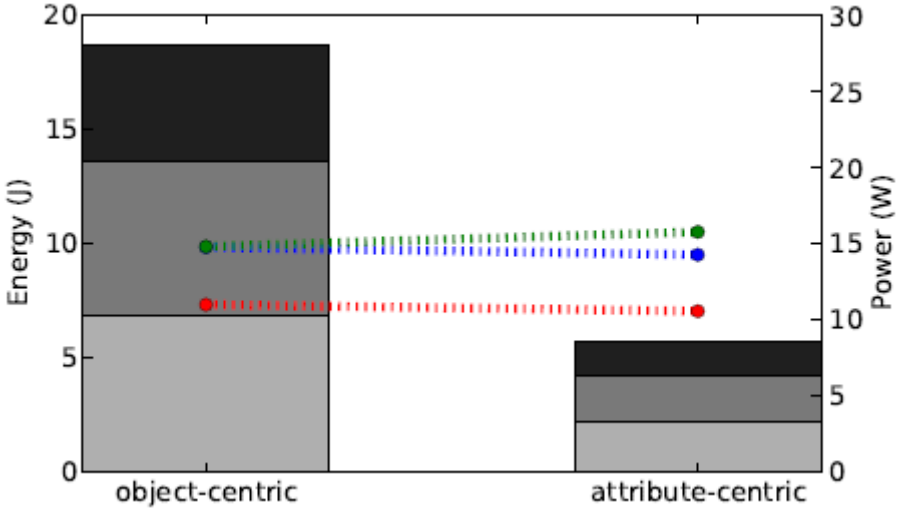
```
class Grouped {
  int a, b, c, d, e = ...;
}
class Main {
  Grouped[] group = ...;
  void calc() {
    for (int i = 0; i < N; i++) {
      group[i].e = group[i].a * group[i].b * group[i].c * group[i].d;
}}}
```

Fig. 1. Object-Centric Data Grouping

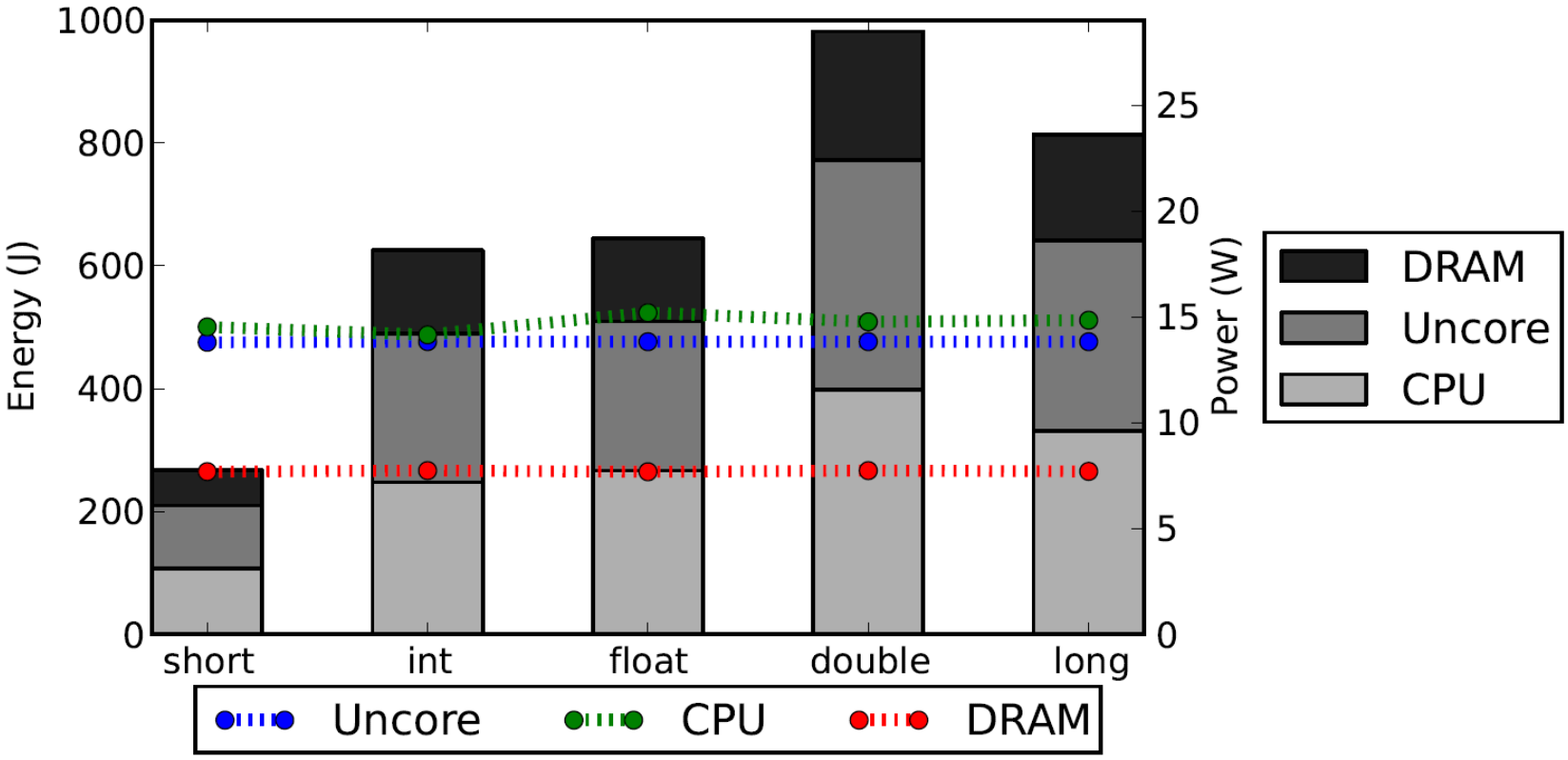
```
class Main {
  int[] a = ..; int[] b = ..; int[] c = ..; int[] d = ..; int[] e = ..;
  void calc() {
    for (int i = 0; i < N; i++) {
      e[i] = a[i] * b[i] * c[i] * d[i];
}}}
```

Fig. 2. Attribute-Centric Data Grouping

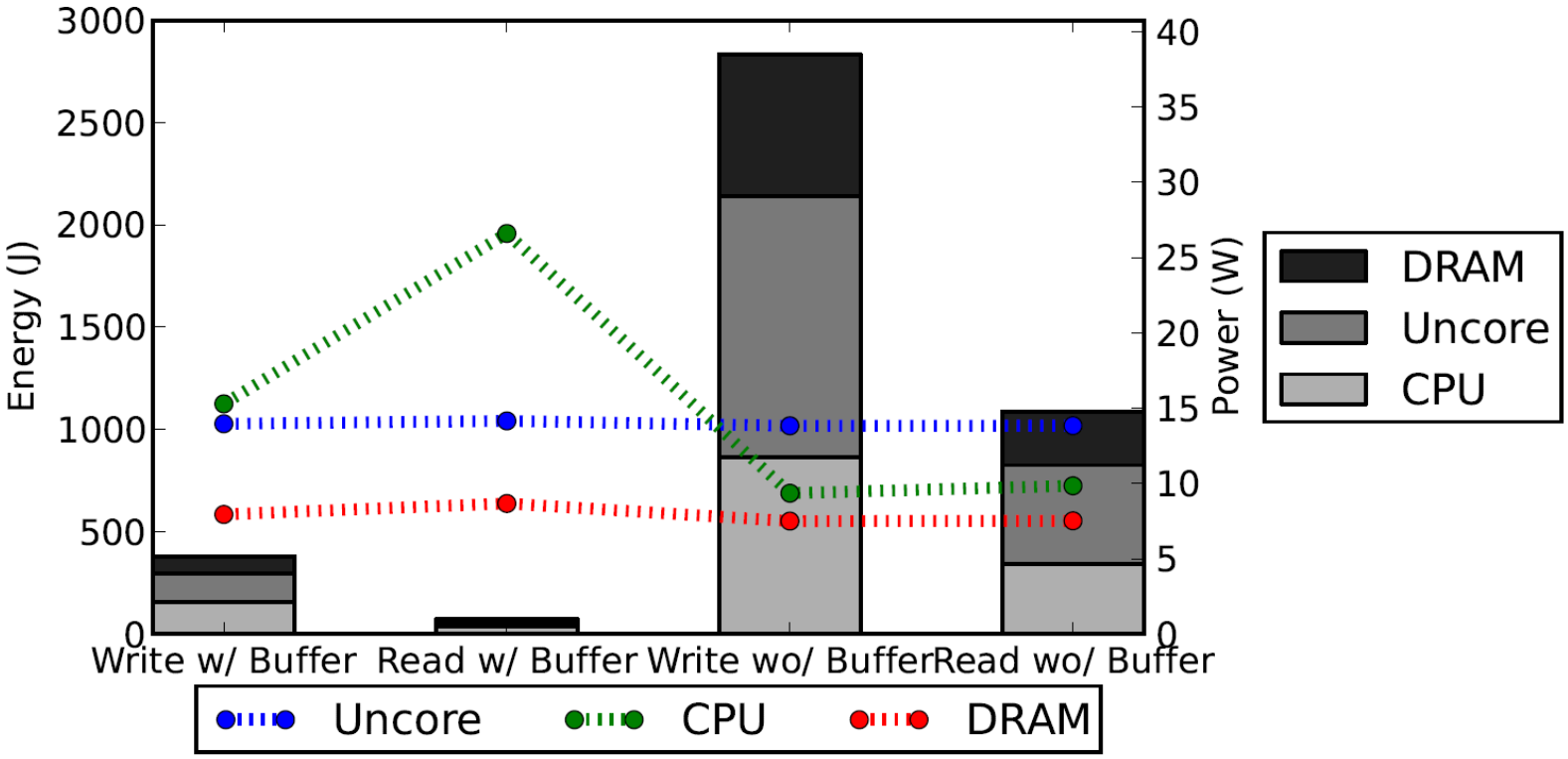
Data Organization



Data Precision Choices



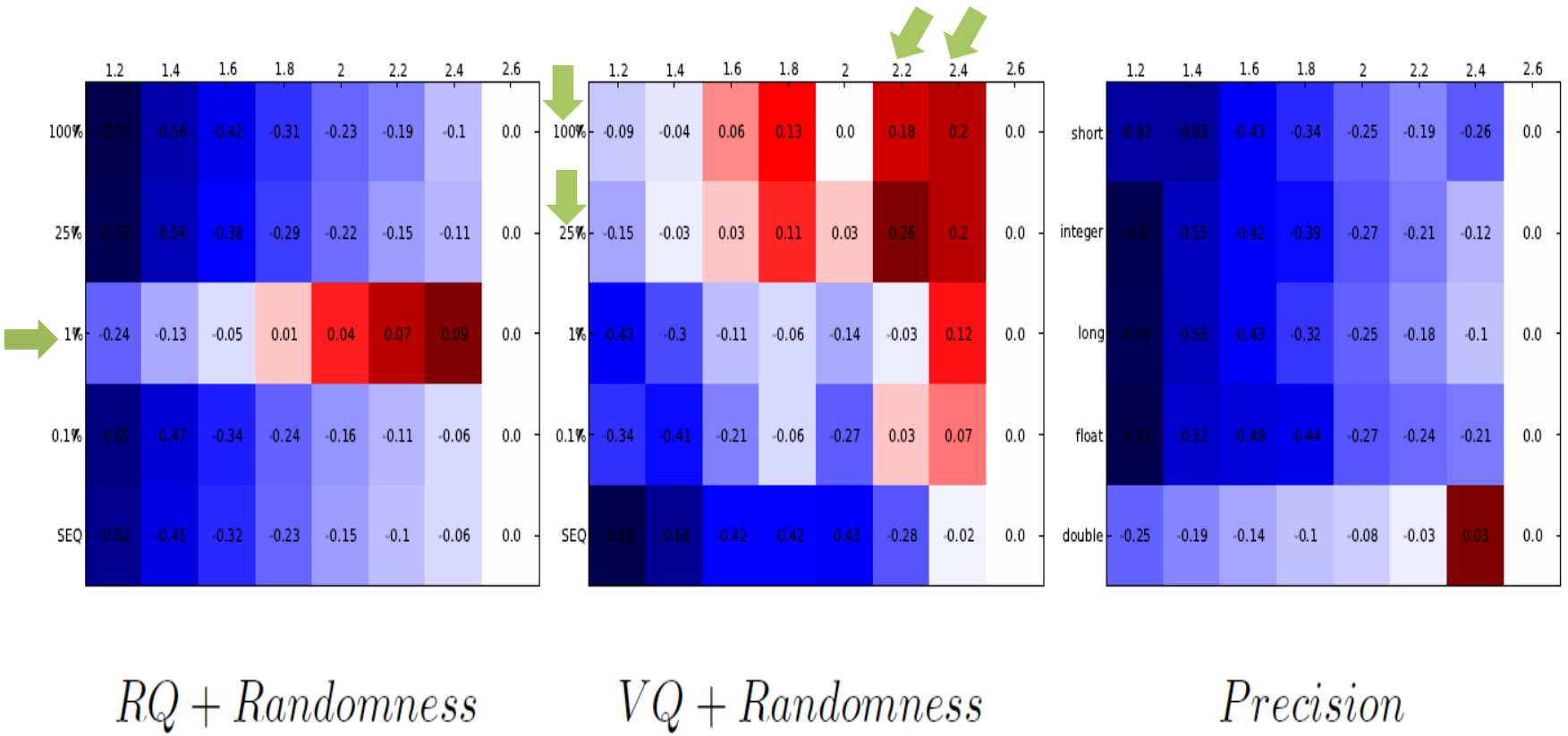
Data I/O Congurations



RQ2 Hardware-Level Interaction

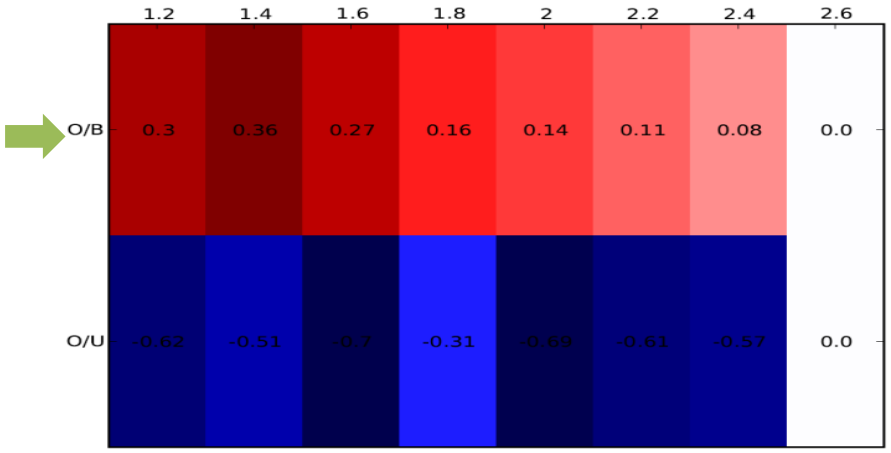
- Investigate RQ1 data-oriented application features in the context of Dynamic Voltage and Frequency Scaling (DVFS)

Unifying Application-Level Optimization with DVFS

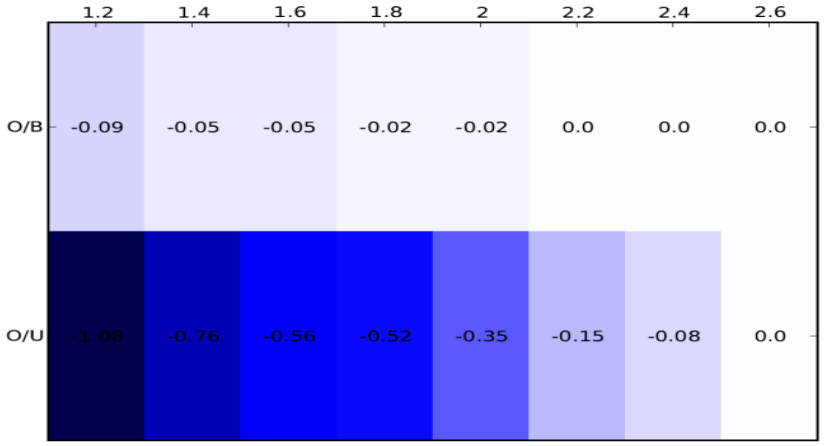


$RQ + Randomness$ $VQ + Randomness$ $Precision$

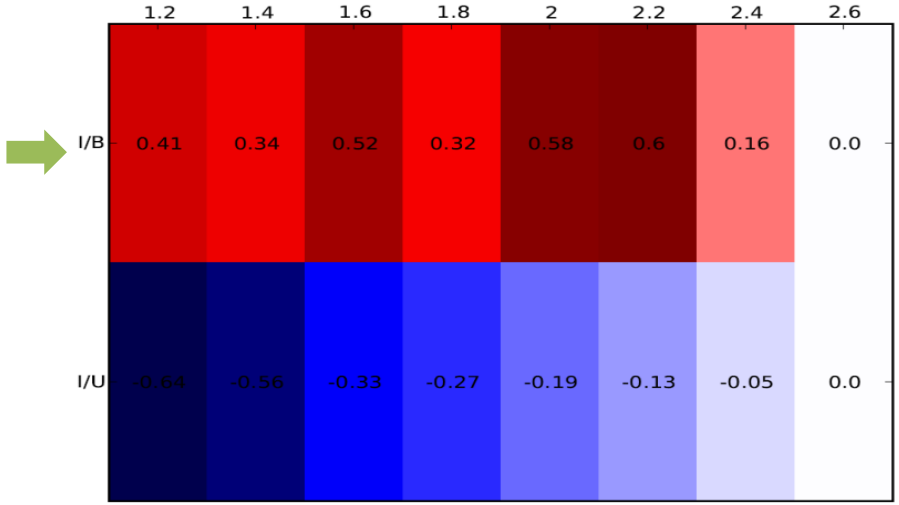
Unifying Application-Level Optimization with DVFS



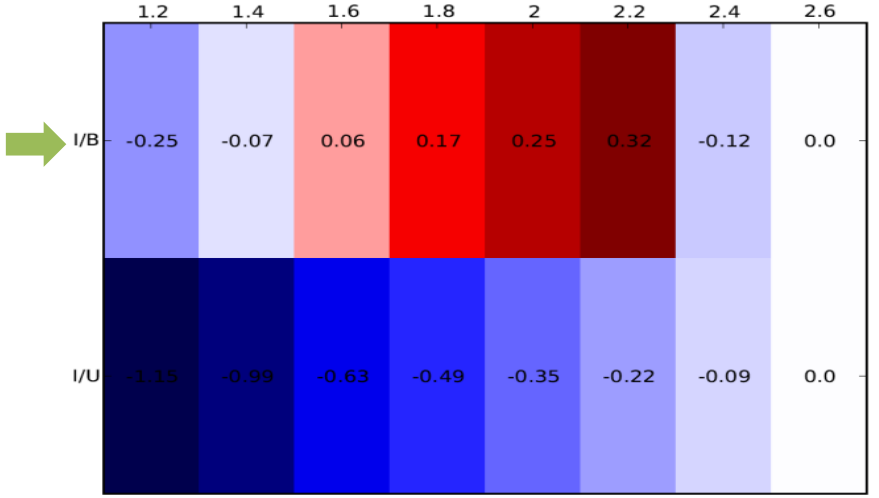
Output Energy



Output Time

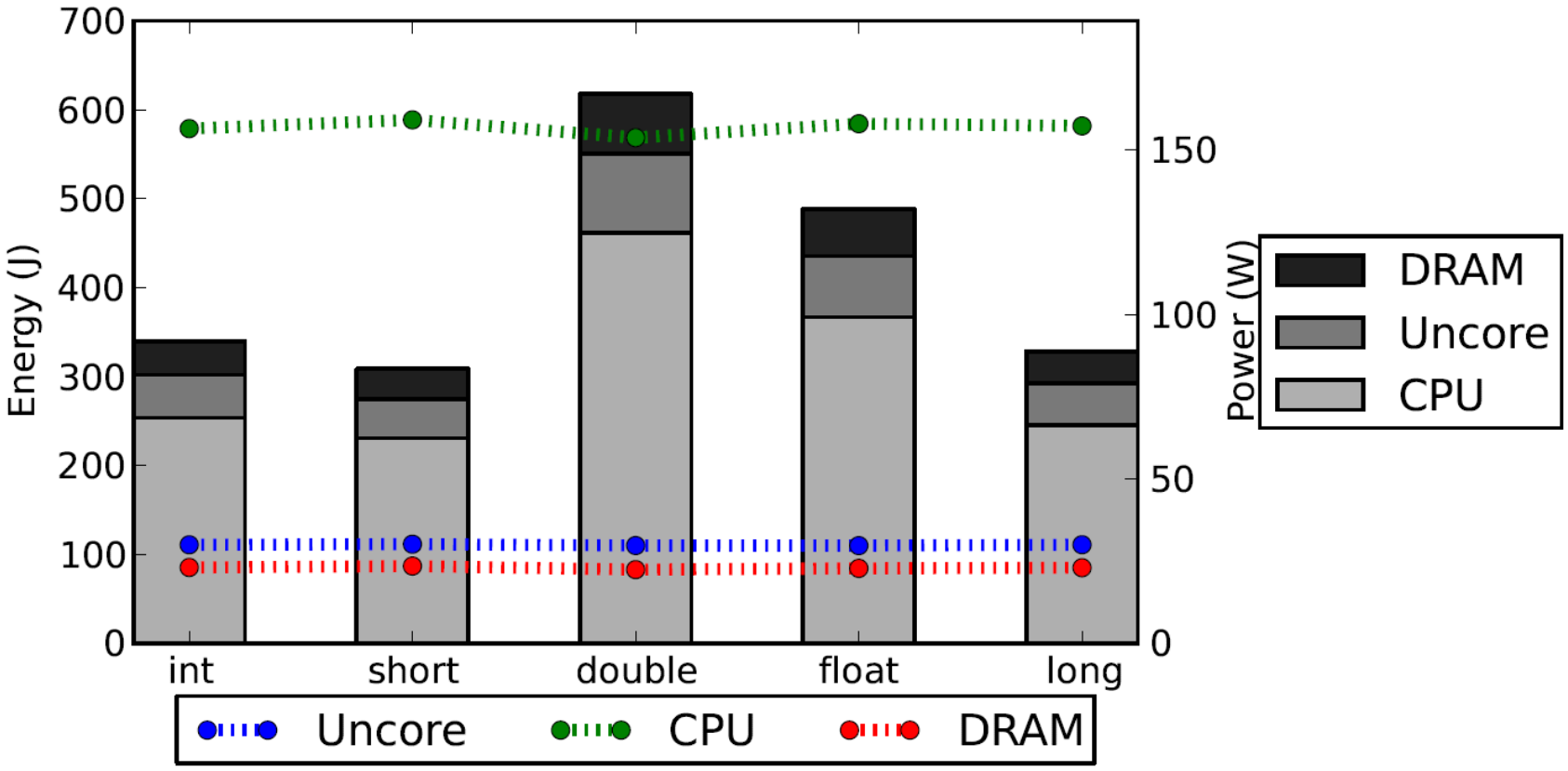


Input Energy



Input Time

Case Study – Sunflow



Conclusions

- It focuses on application-level features, instead of hard-ware performance counters, CPU instructions, or VM bytecode;
- It is carried out from the data-oriented perspective, charting an optimization space often known to be too “application-specic" to quantify and generalize;
- It offers the rest clues on the impact of unifying application-level energy management and hardware-level energy management;
- it provides an in-depth analysis from a whole-system perspective, considering energy consumption not only resulting from CPU cores, but also from caches and DRAM.

THANK YOU!