Performance Comparison for Scientific Computations on the Edge via Relative Performance

Speaker : **Aravind Sankaran**[1]
Supervisor : **Paolo Bientinesi**[2]

3rd Workshop on Parallel AI and Systems for the Edge (PAISE)
May 21, 2021

[1] International Research Training Group for Modern Inverse Problem, RWTH Aachen University, Germany
[2] Department of Computing Science, Umeå Universitet, Sweden

Modern Inverse Problems | RWTH AACHEN UNIVERSITY

Abstract Scientific Code

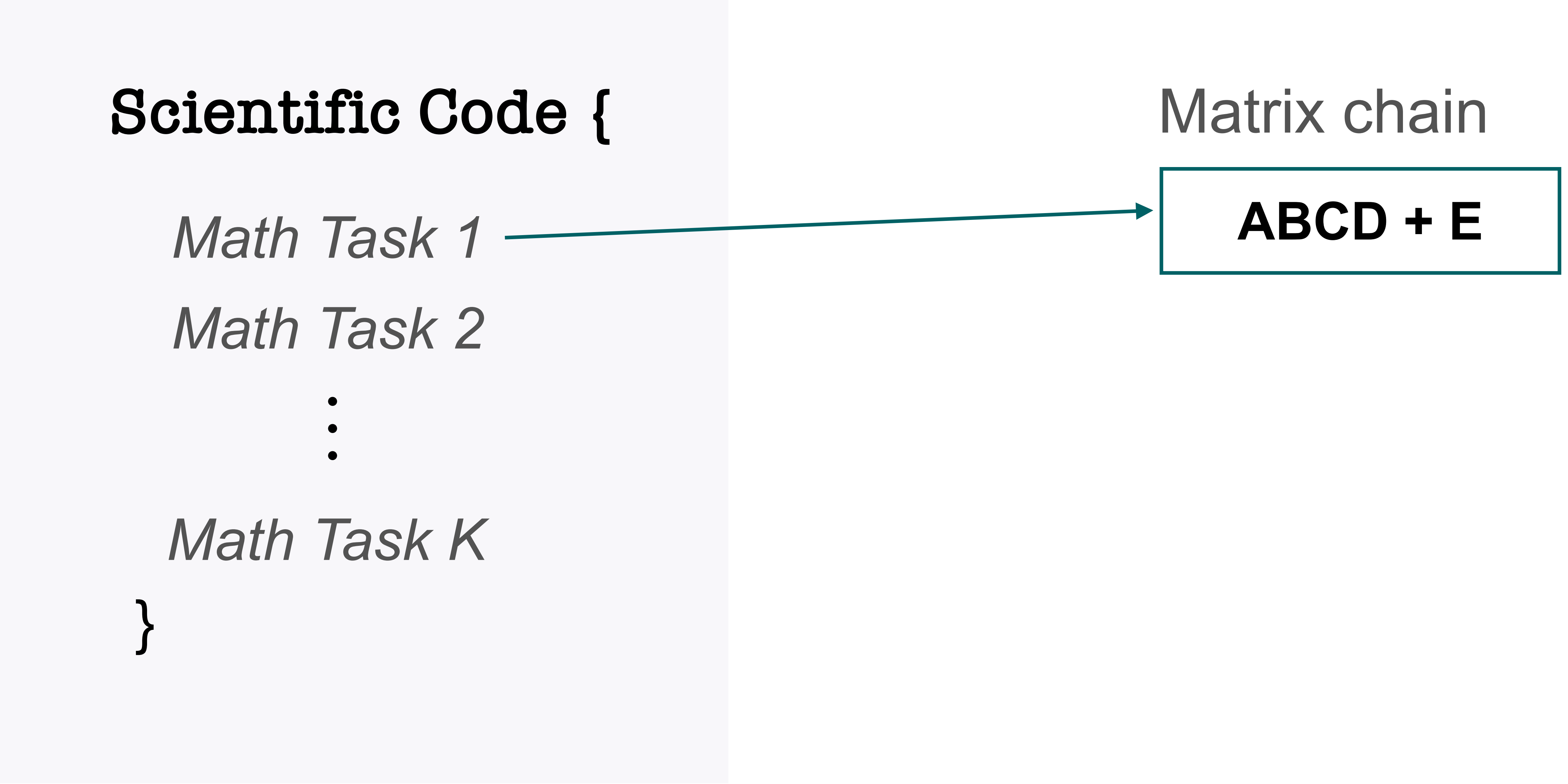
```
Scientific Code {
    Math Task 1
    Math Task 2
    ⋮
    Math Task K
}
```

Setup:

We consider scientific codes consisting of sequence of math tasks.

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Abstract Scientific Code



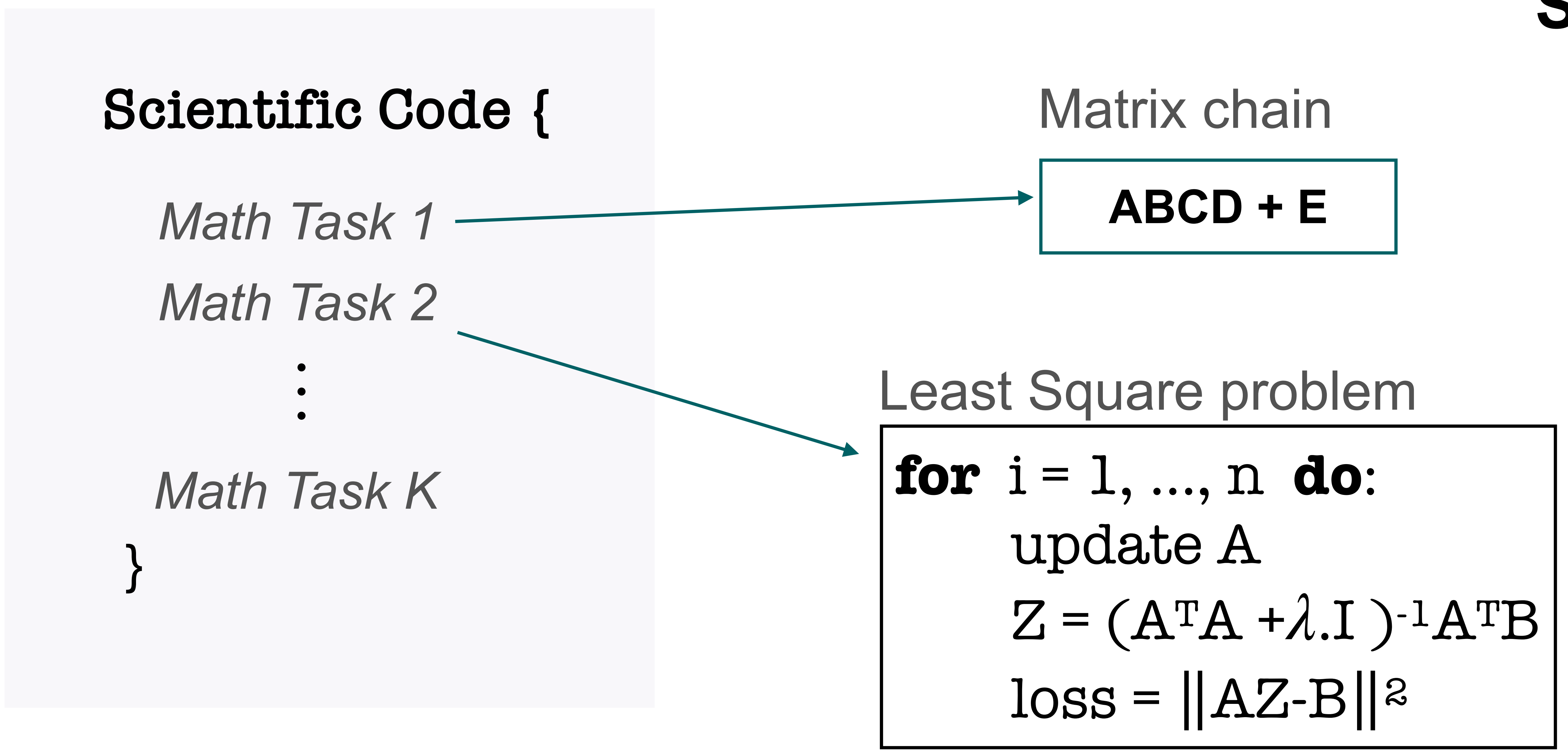
Setup:

We consider scientific codes consisting of sequence of math tasks.

Some examples of math tasks:

1. Linear algebra operations.

Abstract Scientific Code



Setup:

We consider scientific codes consisting of sequence of math tasks.

Some examples of math tasks:

1. Linear algebra operations.
2. Evaluation of some loss function in a loop.

Performance Comparison for Scientific Computations on the Edge via Relative Performance
Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden
PAISE Workshop | May 21, 2021

Abstract Scientific Code

Scientific Code {

Math Task 1

Math Task 2

⋮

Math Task K

}

Matrix chain

ABCD + E

Least Square problem

for $i = 1, \ldots, n$ **do**:

update A

$Z = (A^TA + \lambda.I)^{-1}A^TB$

$loss = ||AZ-B||^2$

Setup:

We consider scientific codes consisting of sequence of math tasks.

Some examples of math tasks:

1. Linear algebra operations.
2. Evaluation of some loss function in a loop.
3. Neural network.

Performance Comparison for Scientific Computations on the Edge via Relative Performance
Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden
PAISE Workshop | May 21, 2021

Partitioning and distribution of Scientific Code among various devices

Scientific Code {

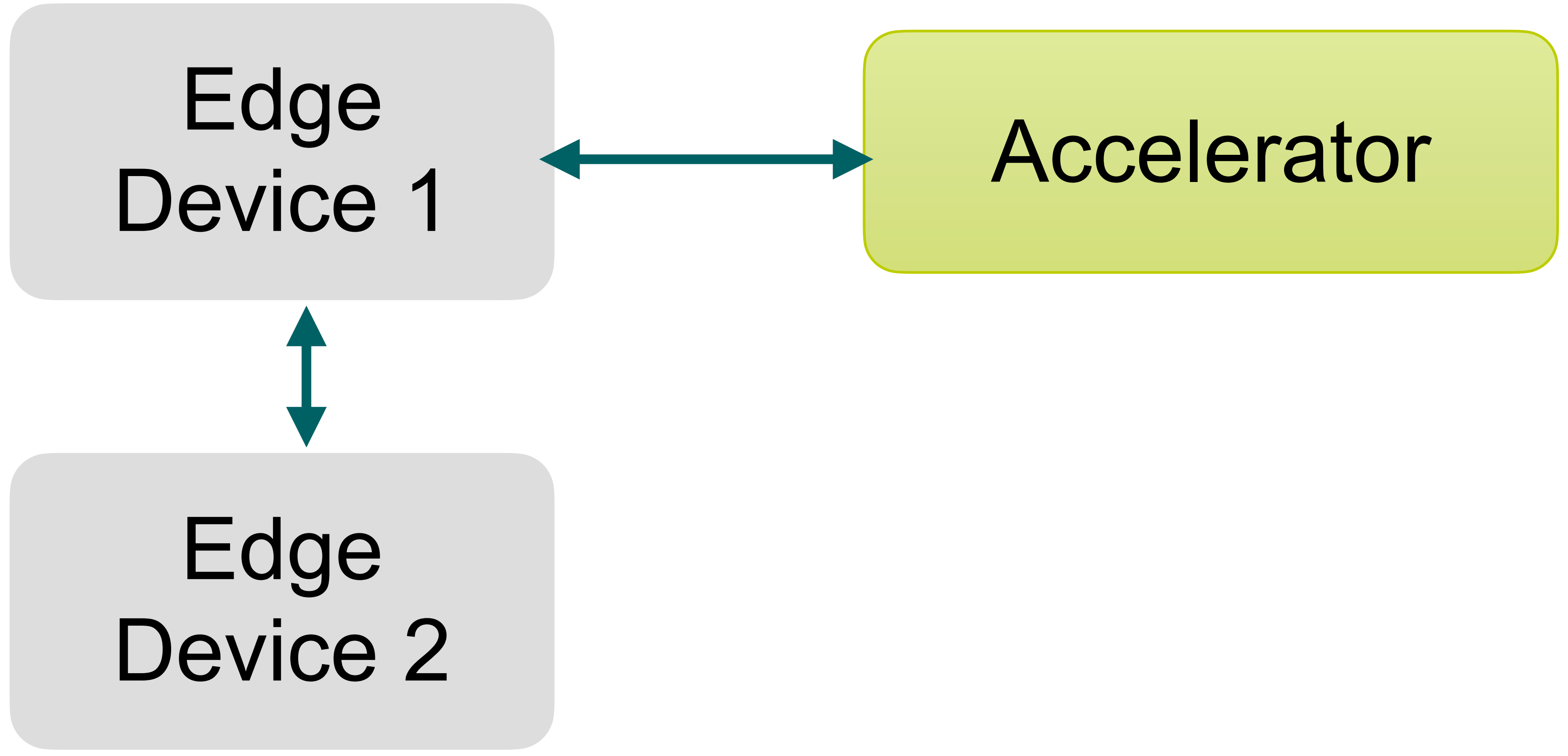
Math Task 1

Math Task 2

Math Task N

}

Computations can be split among devices



Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Partitioning and distribution of Scientific Code among various devices

```
Scientific Code {
    Math Task 1
    Math Task 2

    Math Task N

}
```

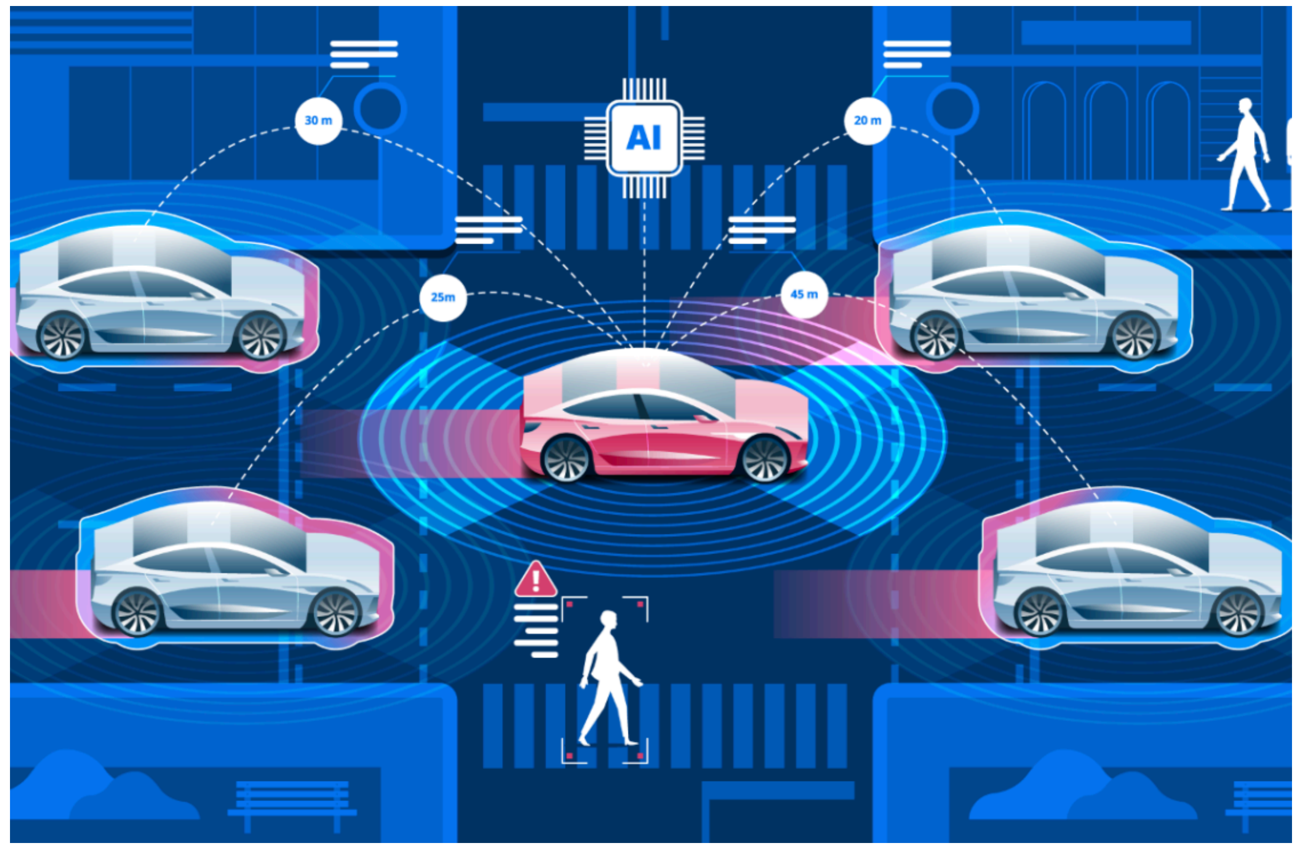
Computations can be split among devices

Applications:

1) Connected Autonomous Vehicles.

Edge Device 1 ↔ Accelerator

Edge Device 1 ↕ Edge Device 2



Source: https://www.smartcitiesworld.net/

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Partitioning and distribution of Scientific Code among various devices

Scientific Code {

Math Task 1

Math Task 2

Math Task N

}

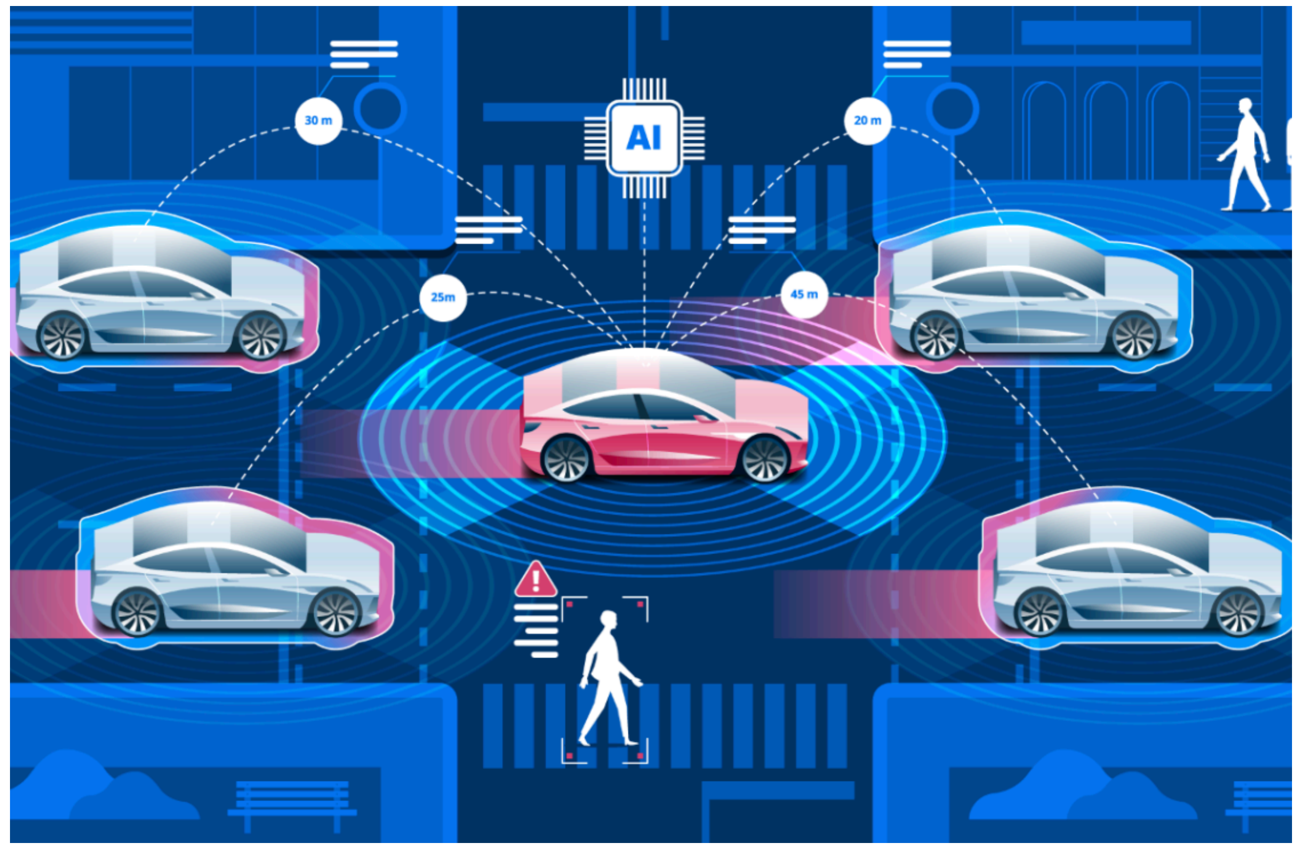
Computations can be split among devices

Edge Device 1 ↔ Accelerator

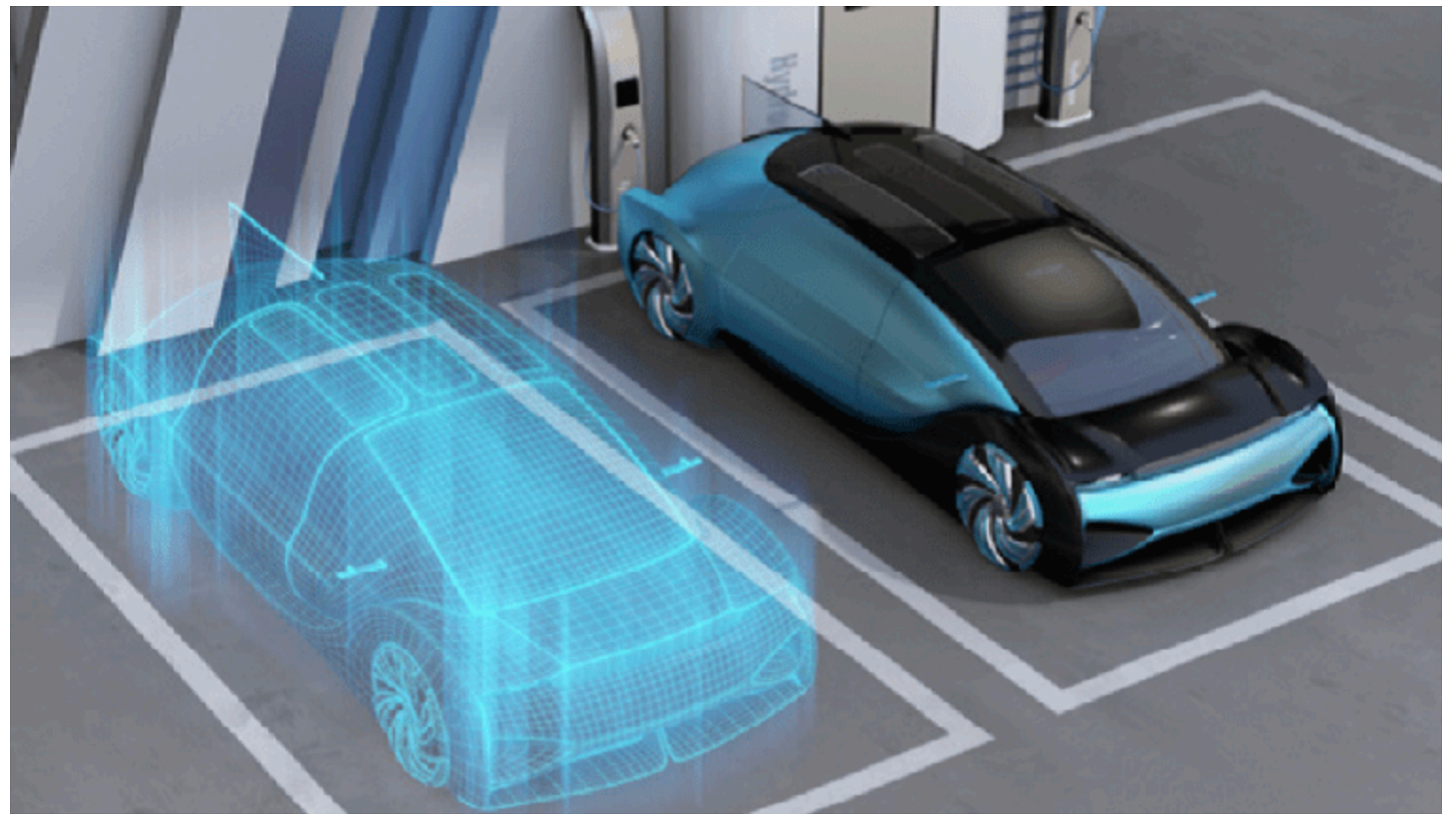
Edge Device 1 ↕ Edge Device 2

Applications:

1) Connected Autonomous Vehicles.
2) Digital Twins.



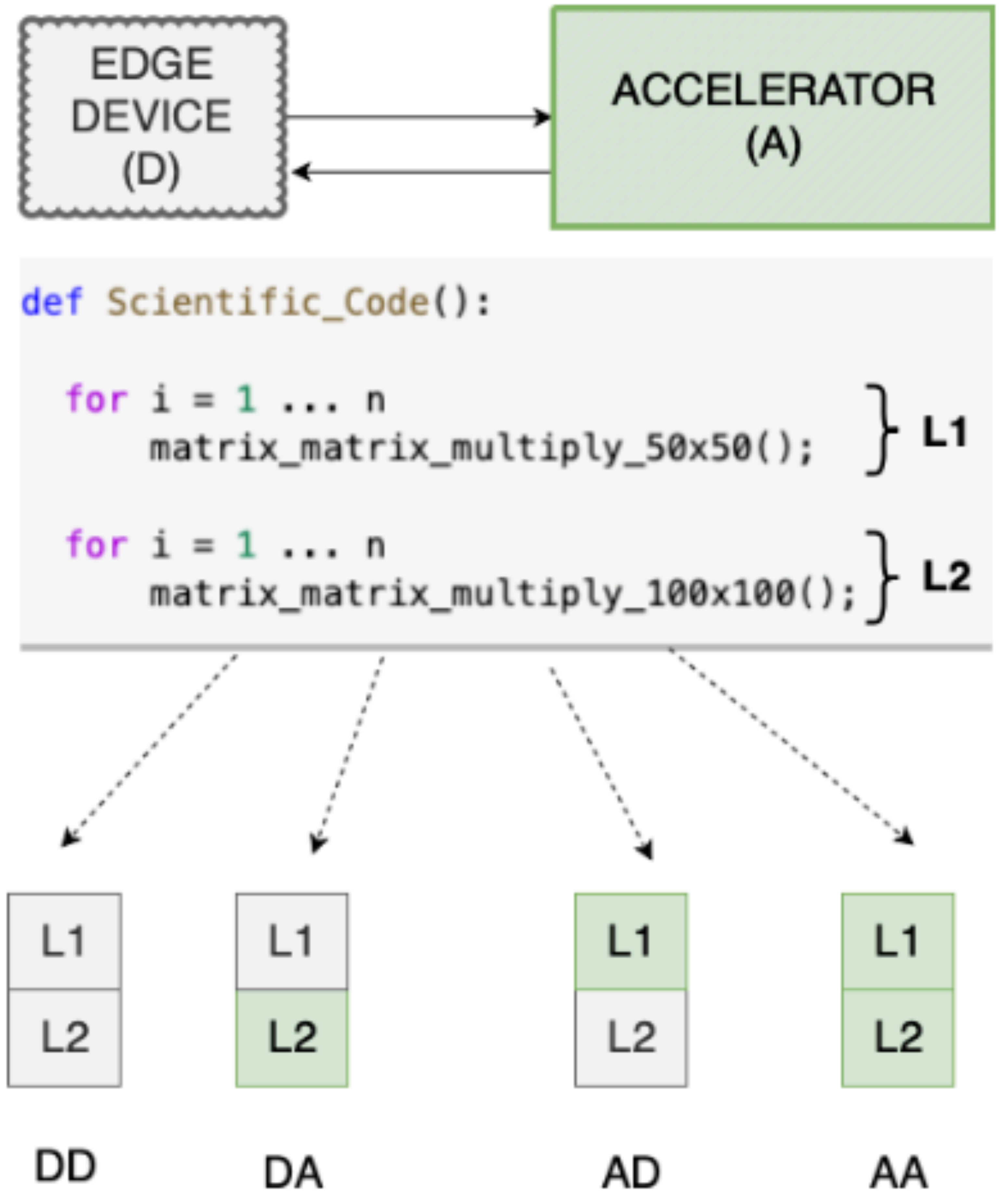
Source: https://www.smartcitiesworld.net/



Source: https://metrology.news

Performance Comparison for Scientific Computations on the Edge via Relative Performance
Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden
PAISE Workshop | May 21, 2021

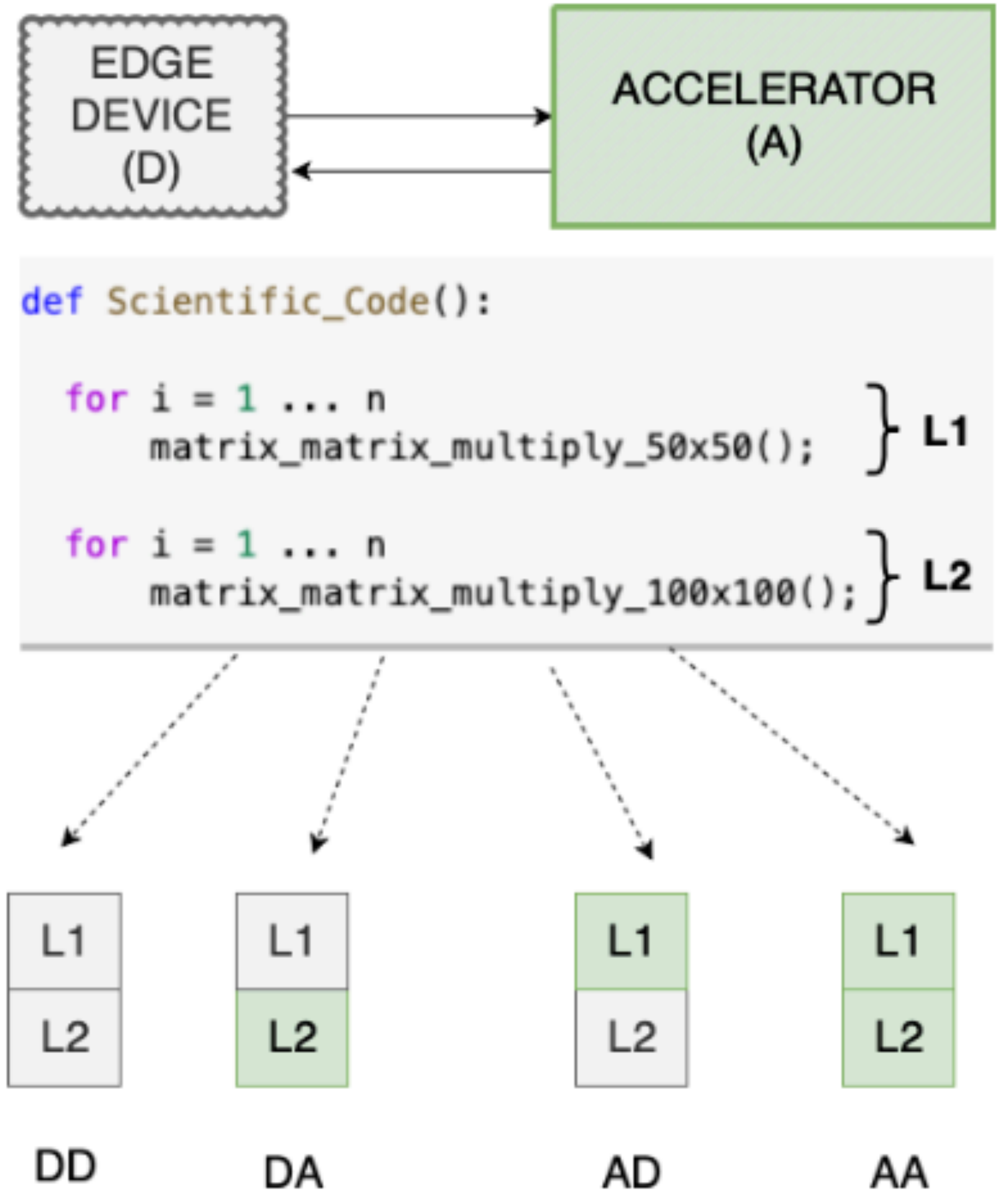
Capturing Performance variations



An example showing four possible ways of splitting a scientific code between CPU and GPU:

To simplify our explanation, we assume L1 and L2 **cannot** be executed concurrently.

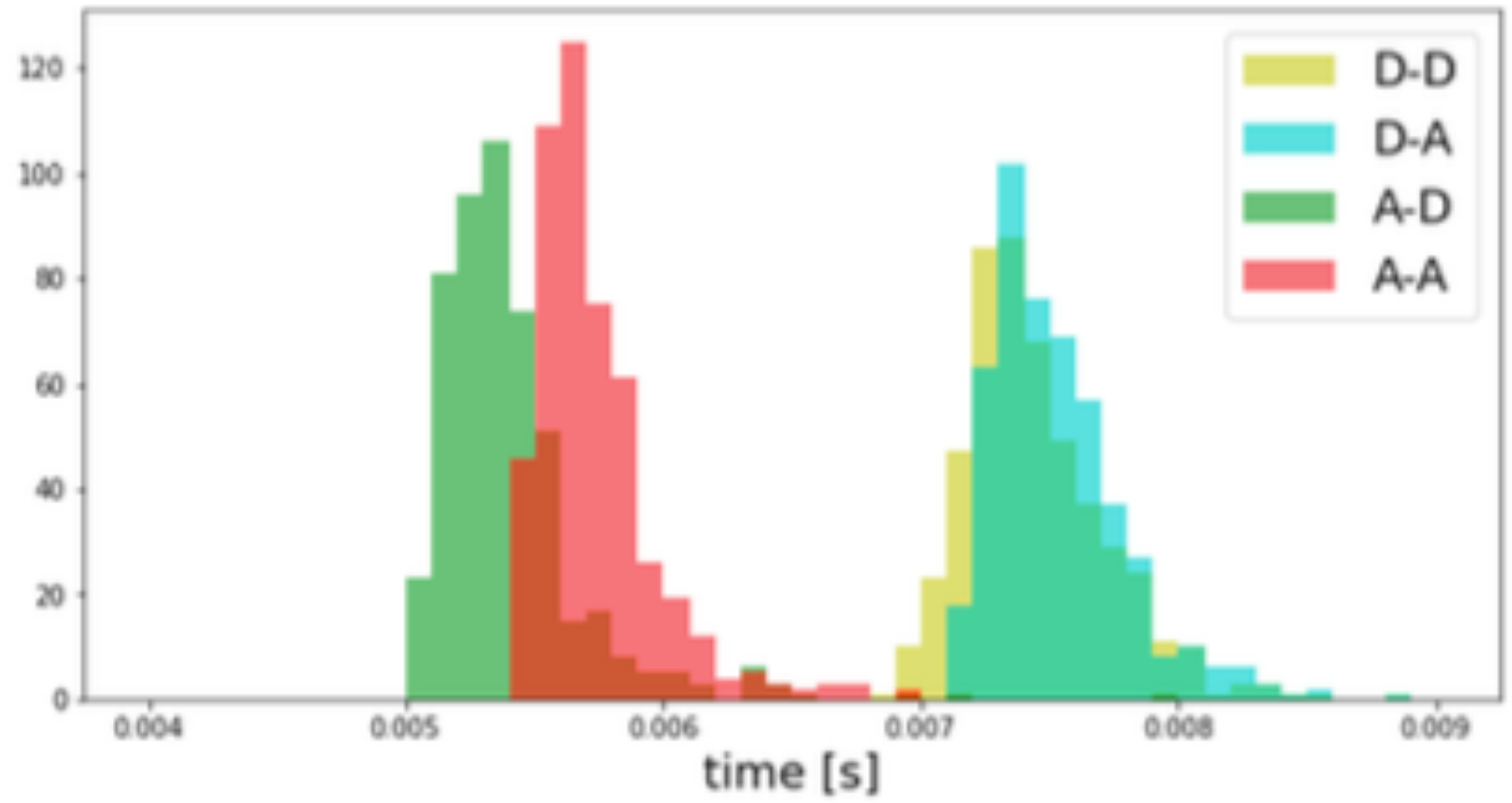
Capturing Performance variations



An example showing four possible ways of splitting a scientific code between CPU and GPU:

We assume L1 and L2 **cannot** be executed concurrently.

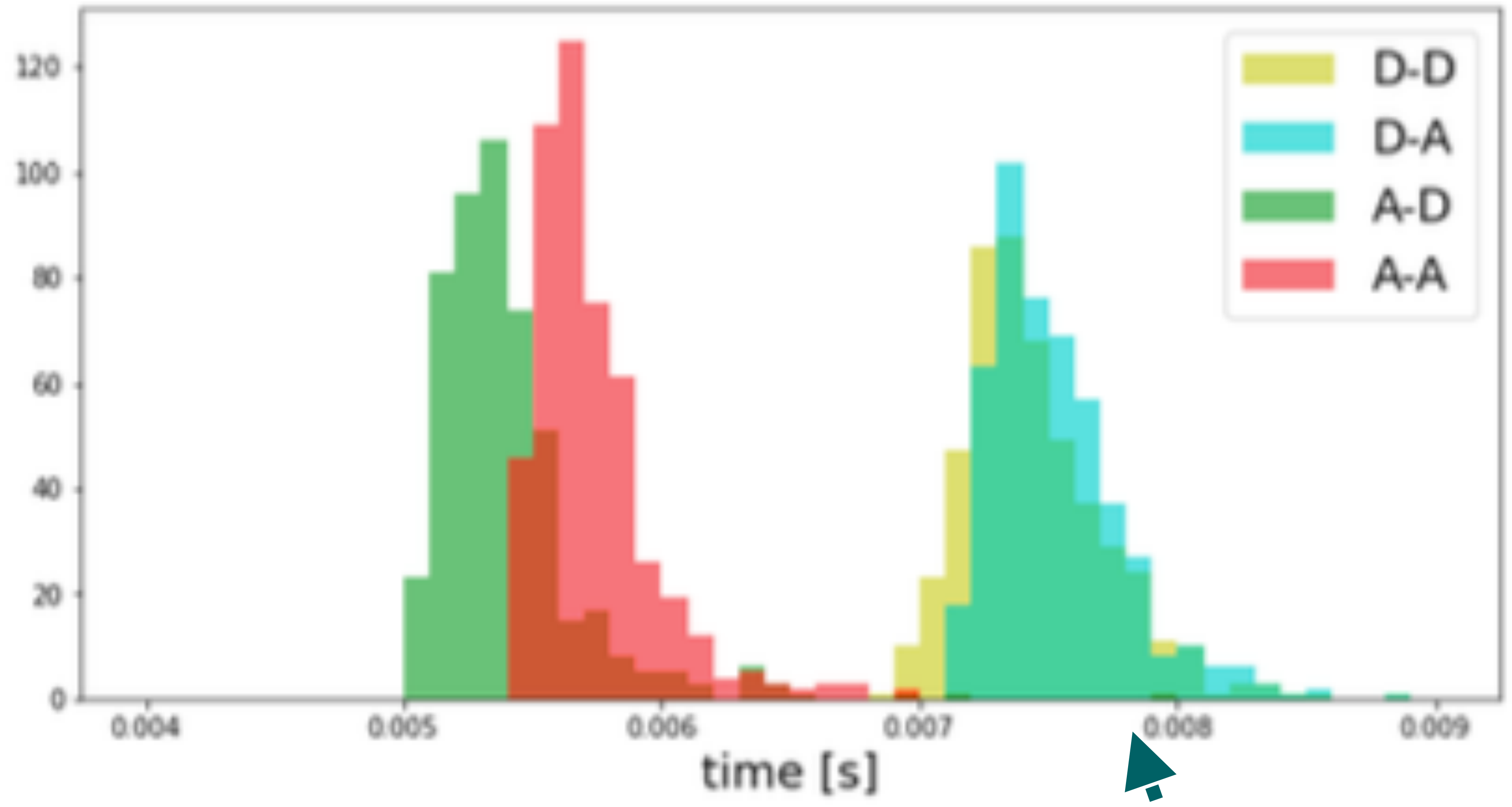
Histogram of execution times indicates differences in performance.



Performance Comparison for Scientific Computations on the Edge via Relative Performance
Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden
PAISE Workshop | May 21, 2021

Clustering into performance classes

Histogram of execution times

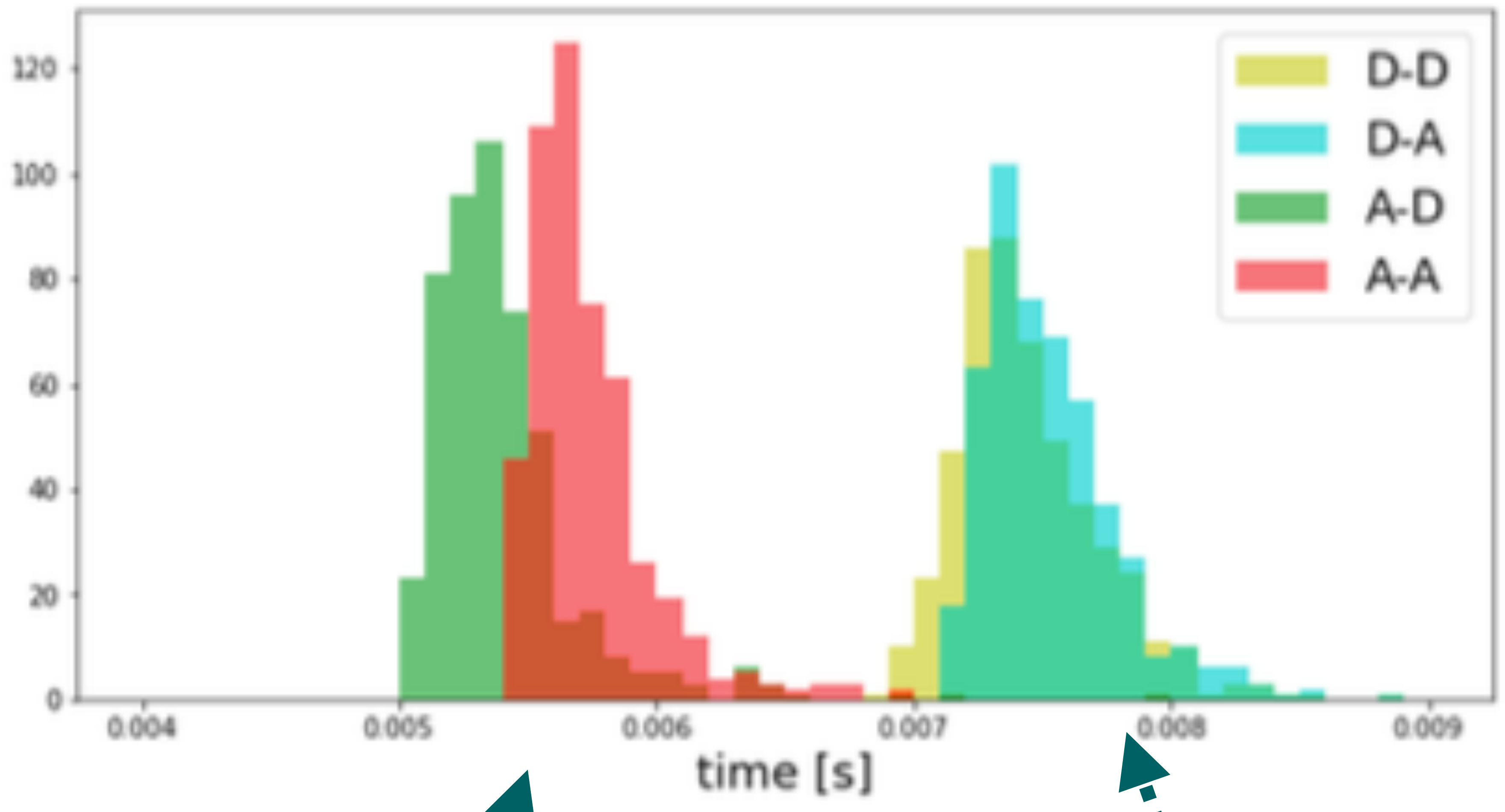


Same cluster

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Clustering into performance classes

Histogram of execution times



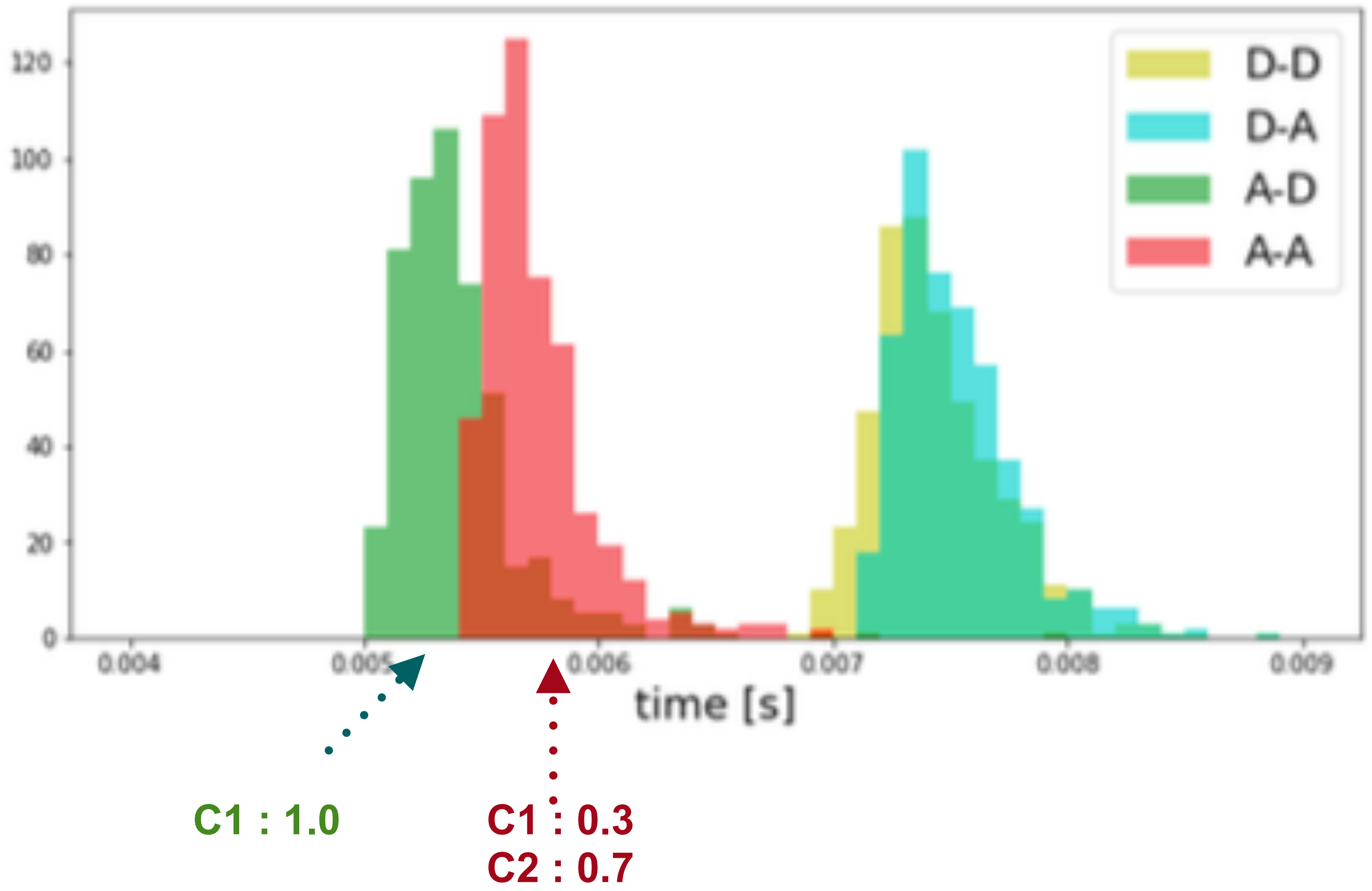
Same cluster?!

Same cluster

Modern Inverse Problems

Clustering into performance classes

Histogram of execution times



Soft clustering

An implementation can be assigned to more than one cluster.

The relative score indicates the confidence of assignment to a particular cluster.

Cluster	*Algorithm*	*Relative Score*
C1	**AD** **AA**	**1.0** **0.3**
C2	**AA** **DD** **DA**	**0.7** **0.3** **0.3**
C3	**DD** **DA**	**0.7** **0.6**
C4	**DA**	**0.1**

Clustering into performance classes

Consider another example with three math tasks:

```
Scientific Code {

    Math Task 1
    Math Task 2
    Math Task 3

}
```

Cluster	Algorithm	Relative Score
$\mathcal{C}_1$	$\mathbf{alg_{DDA}}$	1.0
	$\mathbf{alg_{DAA}}$	0.6
$\mathcal{C}_2$	$\mathbf{alg_{DDD}}$	1.0
	$\mathbf{alg_{DAA}}$	0.4
$\mathcal{C}_3$	$\mathbf{alg_{ADA}}$	1.0
	$\mathbf{alg_{ADD}}$	1.0
	$\mathbf{alg_{DAD}}$	0.7
$\mathcal{C}_4$	$\mathbf{alg_{AAA}}$	1.0
	$\mathbf{alg_{DAD}}$	0.3
$\mathcal{C}_5$	$\mathbf{alg_{AAD}}$	1.0

Clustering into performance classes

Consider the following example with three math tasks:

```
Scientific Code {

    Math Task 1
    Math Task 2
    Math Task 3

}
```

Cluster	Algorithm	Relative Score
$\mathcal{C}_1$	alg_{DDA}	1.0
	alg_{DAA}	0.6
$\mathcal{C}_2$	alg_{DDD}	1.0
	alg_{DAA}	0.4
$\mathcal{C}_3$	alg_{ADA}	1.0
	alg_{ADD}	1.0
	alg_{DAD}	0.7
$\mathcal{C}_4$	alg_{AAA}	1.0
	alg_{DAD}	0.3
$\mathcal{C}_5$	alg_{AAD}	1.0

Applications:

1) Energy optimization.

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Clustering into performance classes

Consider the following example with three math tasks:

```
Scientific Code {
    Math Task 1
    Math Task 2
    Math Task 3
}
```

Cluster	Algorithm	Relative Score
$\mathcal{C}_1$	alg_{DDA}	1.0
	alg_{DAA}	0.6
$\mathcal{C}_2$	alg_{DDD}	1.0
	alg_{DAA}	0.4
$\mathcal{C}_3$	alg_{ADA}	1.0
	alg_{ADD}	1.0
	alg_{DAD}	0.7
$\mathcal{C}_4$	alg_{AAA}	1.0
	alg_{DAD}	0.3
$\mathcal{C}_5$	alg_{AAD}	1.0

Applications:

1) Energy optimization.
2) Chart fault tolerant policies.

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Clustering into performance classes

Consider the following example with three math tasks:

```
Scientific Code {

    Math Task 1
    Math Task 2
    Math Task 3

}
```

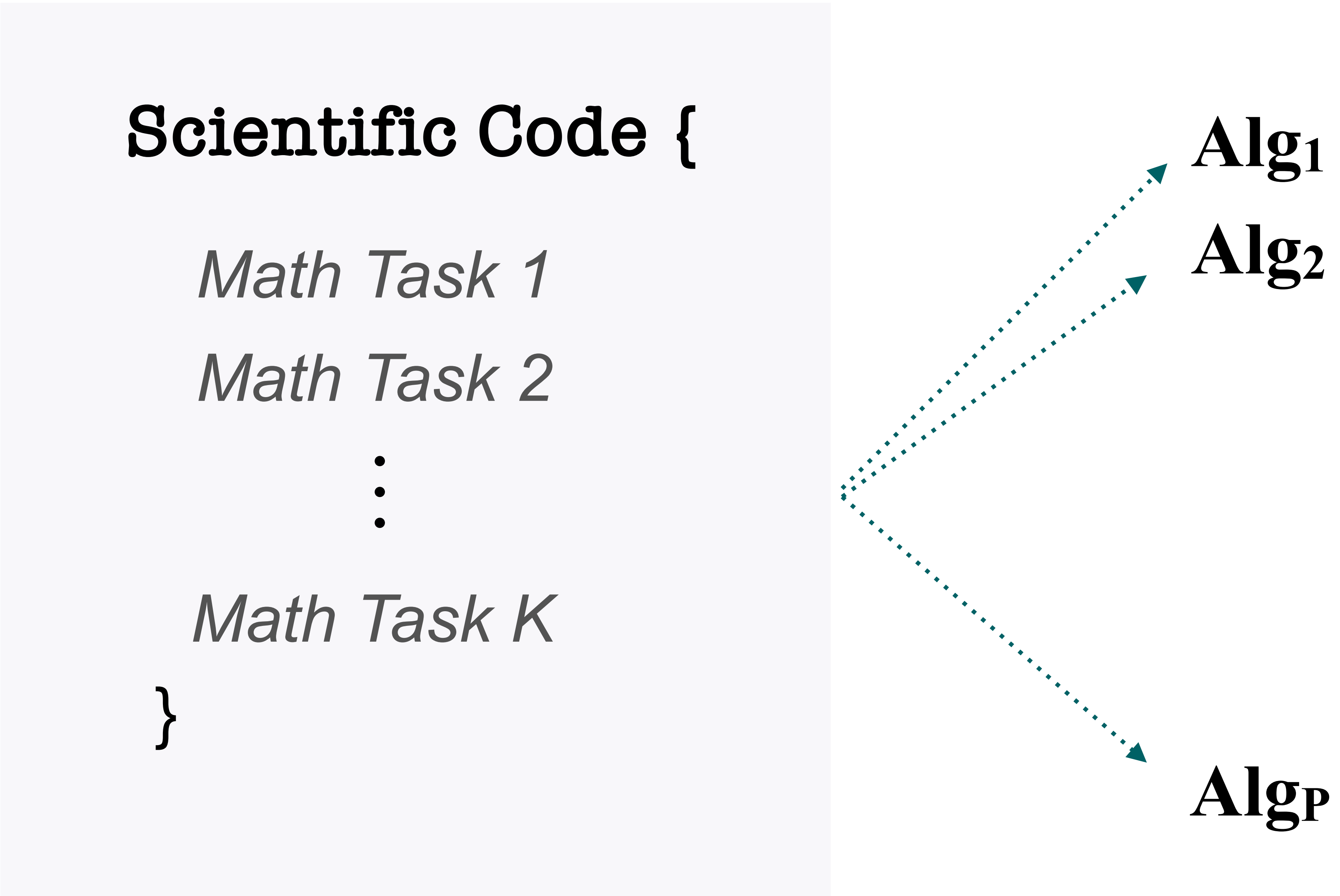
Cluster	Algorithm	Relative Score
$\mathcal{C}_1$	$\mathbf{alg}_{\mathbf{DDA}}$	1.0
	$\mathbf{alg}_{\mathbf{DAA}}$	0.6
$\mathcal{C}_2$	$\mathbf{alg}_{\mathbf{DDD}}$	1.0
	$\mathbf{alg}_{\mathbf{DAA}}$	0.4
$\mathcal{C}_3$	$\mathbf{alg}_{\mathbf{ADA}}$	1.0
	$\mathbf{alg}_{\mathbf{ADD}}$	1.0
	$\mathbf{alg}_{\mathbf{DAD}}$	0.7
$\mathcal{C}_4$	$\mathbf{alg}_{\mathbf{AAA}}$	1.0
	$\mathbf{alg}_{\mathbf{DAD}}$	0.3
$\mathcal{C}_5$	$\mathbf{alg}_{\mathbf{AAD}}$	1.0

Applications:

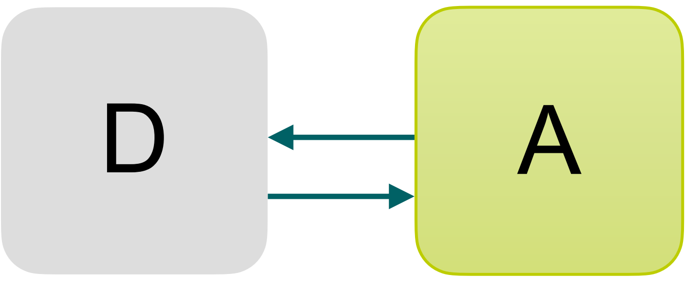
1) Energy optimization.
2) Chart fault tolerant policies.
3) Derive performance models for automatic clustering.

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Clustering Methodology



Let Alg_1 , Alg_2 ... Alg_p be different implementations of the **Scientific Code**



Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Clustering Methodology

Scientific Code {

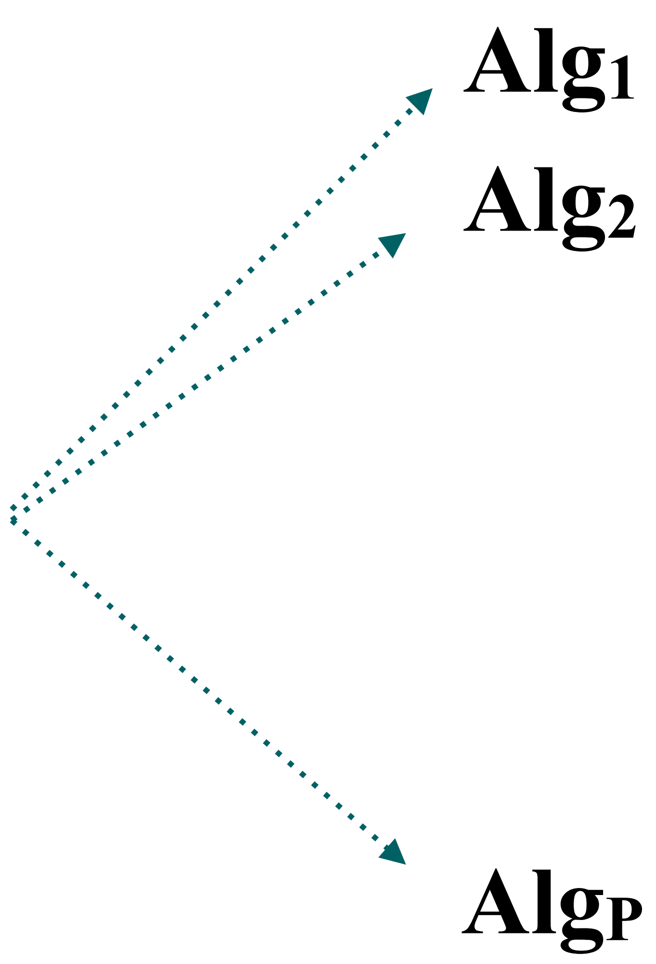
Math Task 1

Math Task 2

⋮

Math Task K

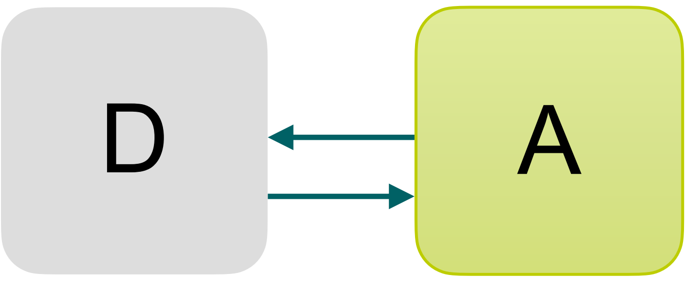
}



Let Alg_1 , Alg_2 ... Alg_p be different implementations of the **Scientific Code**

Steps:

1) Measure each algorithm N times



Clustering Methodology

Scientific Code {

Math Task 1

Math Task 2

⋮

Math Task K

}

Alg_1

Alg_2

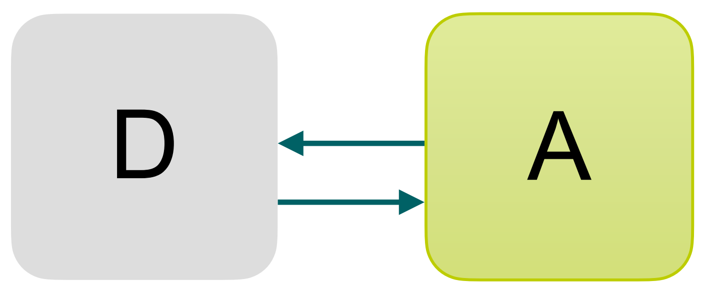
Alg_P

Steps:

1) Measure each algorithm N times

Ensure measurements are invariant system noise.

Alg_1	Alg_2	Alg_3
[oooooooooo	oooooooooo	oooooooooo]



Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Clustering Methodology

Scientific Code {

Math Task 1

Math Task 2

⋮

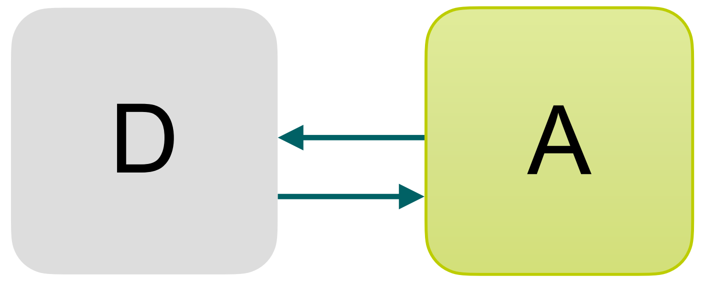
Math Task K

}

Alg_1

Alg_2

Alg_P

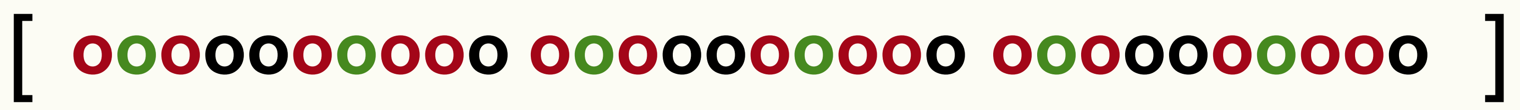


Steps:

1) Measure each algorithm N times

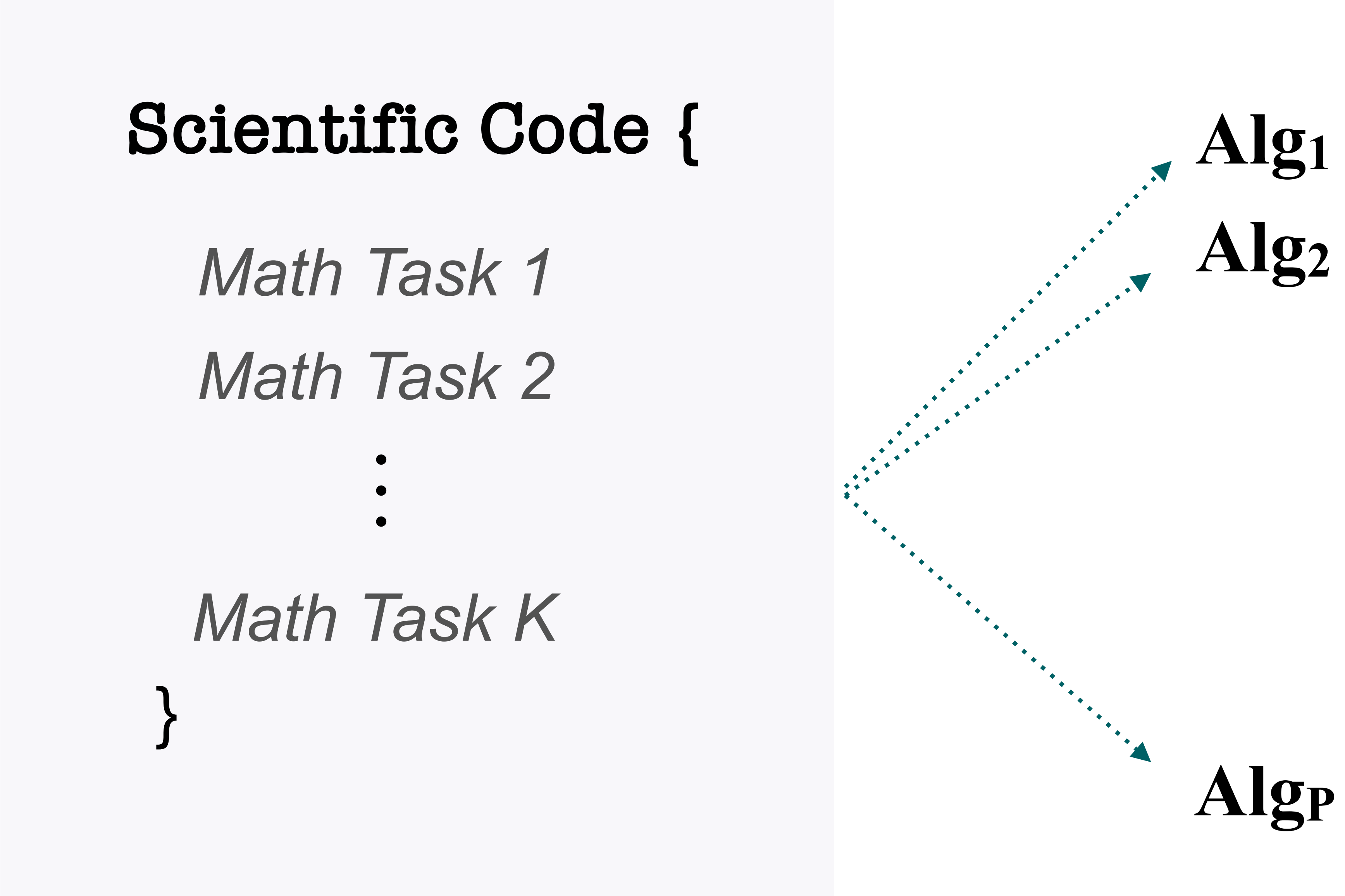
Ensure measurements are invariant system noise.

Alg_1 **Alg_2** **Alg_3**



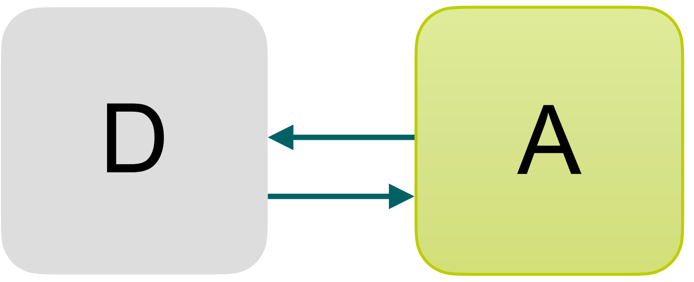
Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Clustering Methodology



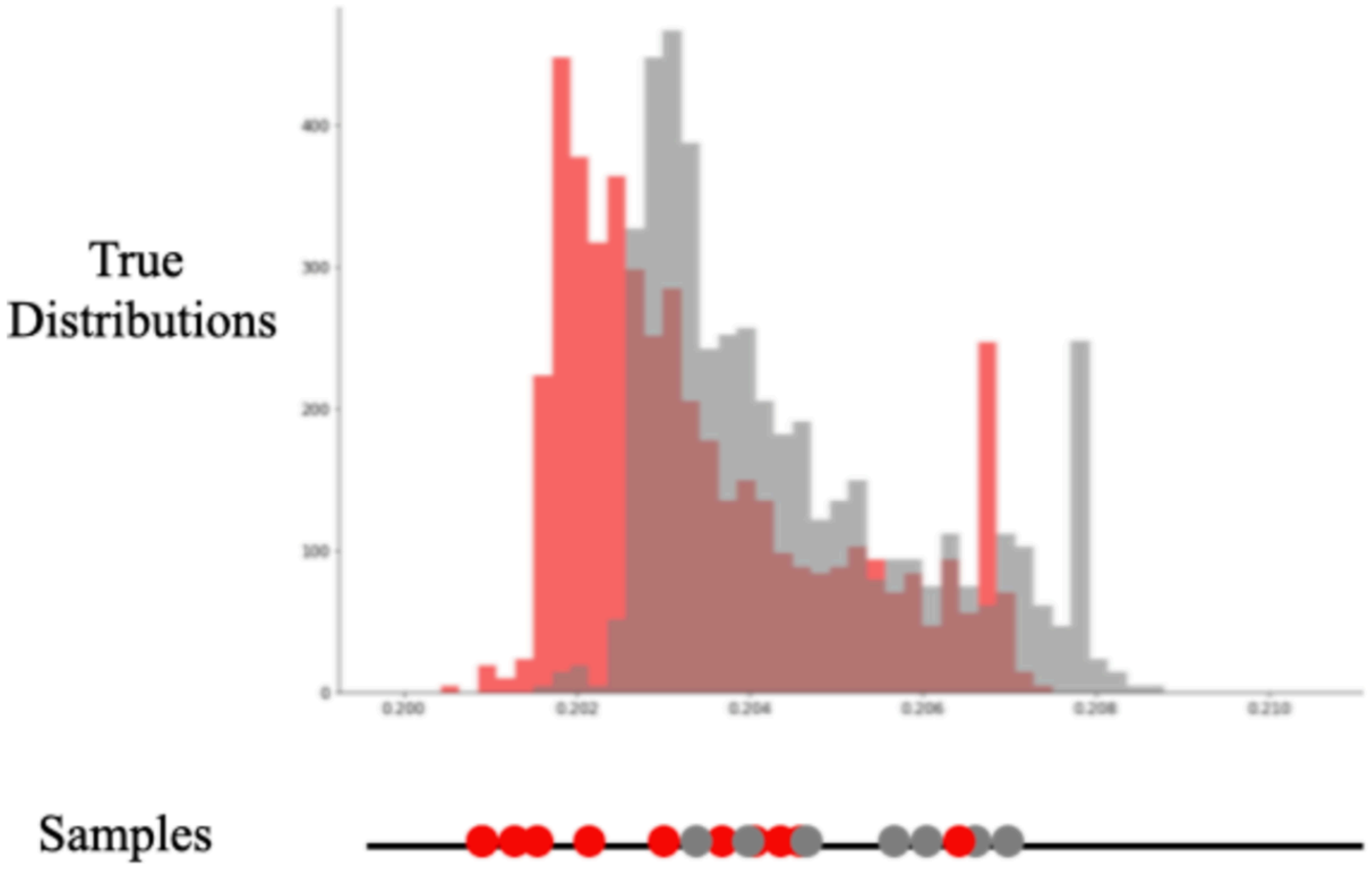
Steps:

1) Measure each algorithm N times.
2) Compare adjacent pairs of algorithms and sort them.



Clustering Methodology

Comparing two algorithms:



True distributions of the algorithms are not available.

We have just a snapshot of the true distribution.

Clustering Methodology

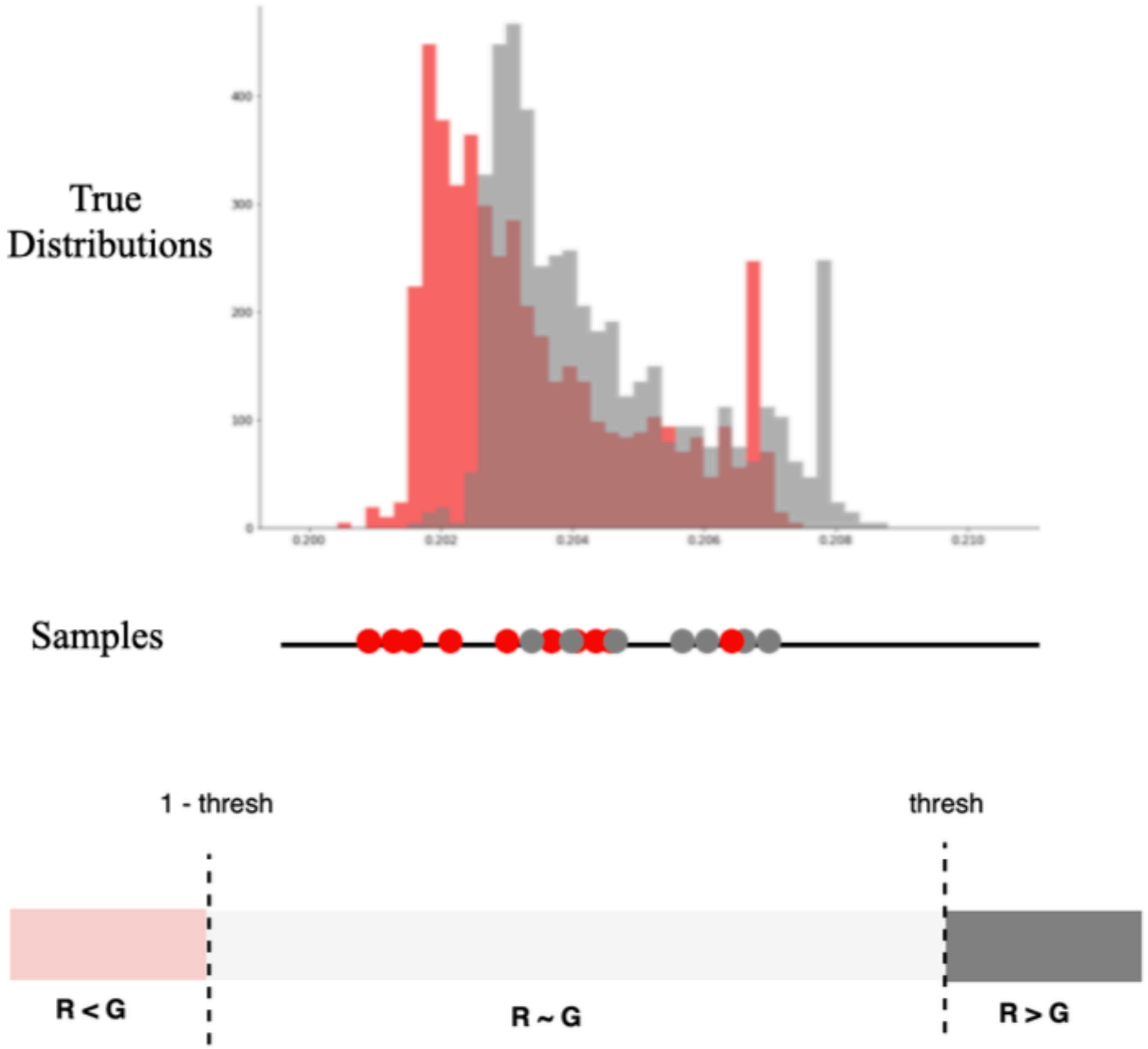
Comparing two algorithms by bootstrapping Distribution Statistics :

$$\mathbf{R} = \{tR_1, tR_2, ..., tR_n\}$$
$$\mathbf{G} = \{tG_1, tG_2, ..., tG_n\}$$

$$\text{Compare}(\mathbf{R}, \mathbf{G}) \in \{<, >, \sim\}$$

Sample $R^S \subset \mathbf{R}$, $G^S \subset \mathbf{G}$

$$\text{Compare}(\mathbf{R}, \mathbf{G}) = \begin{cases} \mathbf{R} > \mathbf{G} & \text{if } \frac{\#(stat(R^s) > stat(G^s))}{M} > thresh \\ \mathbf{R} < \mathbf{G} & \text{if } \frac{\#(stat(R^s) > stat(G^s))}{M} < 1 - thresh \\ \mathbf{R} \sim \mathbf{G} & \text{otherwise} \end{cases}$$



Modern Inverse Problems

RWTH AACHEN UNIVERSITY

Clustering Methodology

Sorting the algorithms using the three-way comparison:

1) Initialise algorithms with consecutive ranks.

2) Compare adjacent pairs of algorithms and swap their ranks if an algorithm occurring later in the sequence performs better than the one occurring earlier.

3) Merge ranks if the two algorithms are performance equivalent.

Algorithm	DD	AA	DA	AD
Rank	1	2	3	4
	"DD" < "AA"			

Algorithm	AA	DD	DA	AD
Rank	1	2	3	4
		"DD" ~ "DA"		

Algorithm	AA	DD	DA	AD
Rank	1	2	2	3
			"DA" < "AD"	

Algorithm	AA	DD	AD	DA
Rank	1	2	2	2
		"DD" < "AD"		

Algorithm	AA	AD	DD	DA
Rank	1	2	3	3
	"AA" < "AD"			

Algorithm	AD	AA	DD	DA
Final Rank	1	2	3	3

Clustering Methodology

Sorting the algorithms using the three-way comparison:

1) Initialise algorithms with consecutive ranks.
2) Compare adjacent pairs of algorithms and sort them.
3) Merge ranks if the two algorithms are performance equivalent.

Algorithm	DD	AA	DA	AD
Rank	1	2	3	4
	"DD" < "AA"			

Algorithm	AA	DD	DA	AD
Rank	1	2	3	4
		"DD" ~ "DA"		

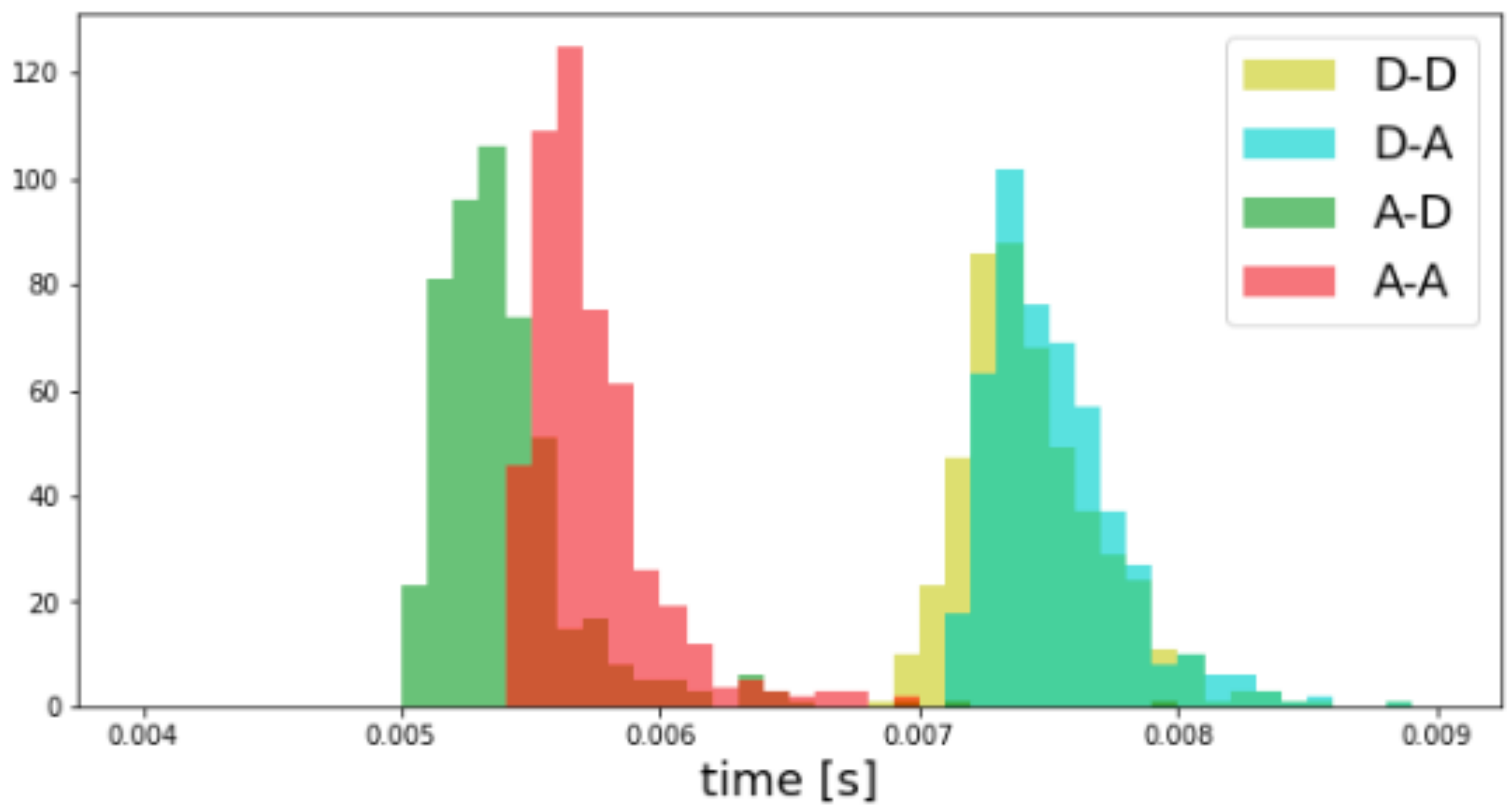
Algorithm	AA	DD	DA	AD
Rank	1	2	2	3
			"DA" < "AD"	

Algorithm	AA	DD	AD	DA
Rank	1	2	2	2
		"DD" < "AD"		

Algorithm	AA	AD	DD	DA
Rank	1	2	3	3
	"AA" < "AD"			

Algorithm	AD	AA	DD	DA
Final Rank	1	2	3	3

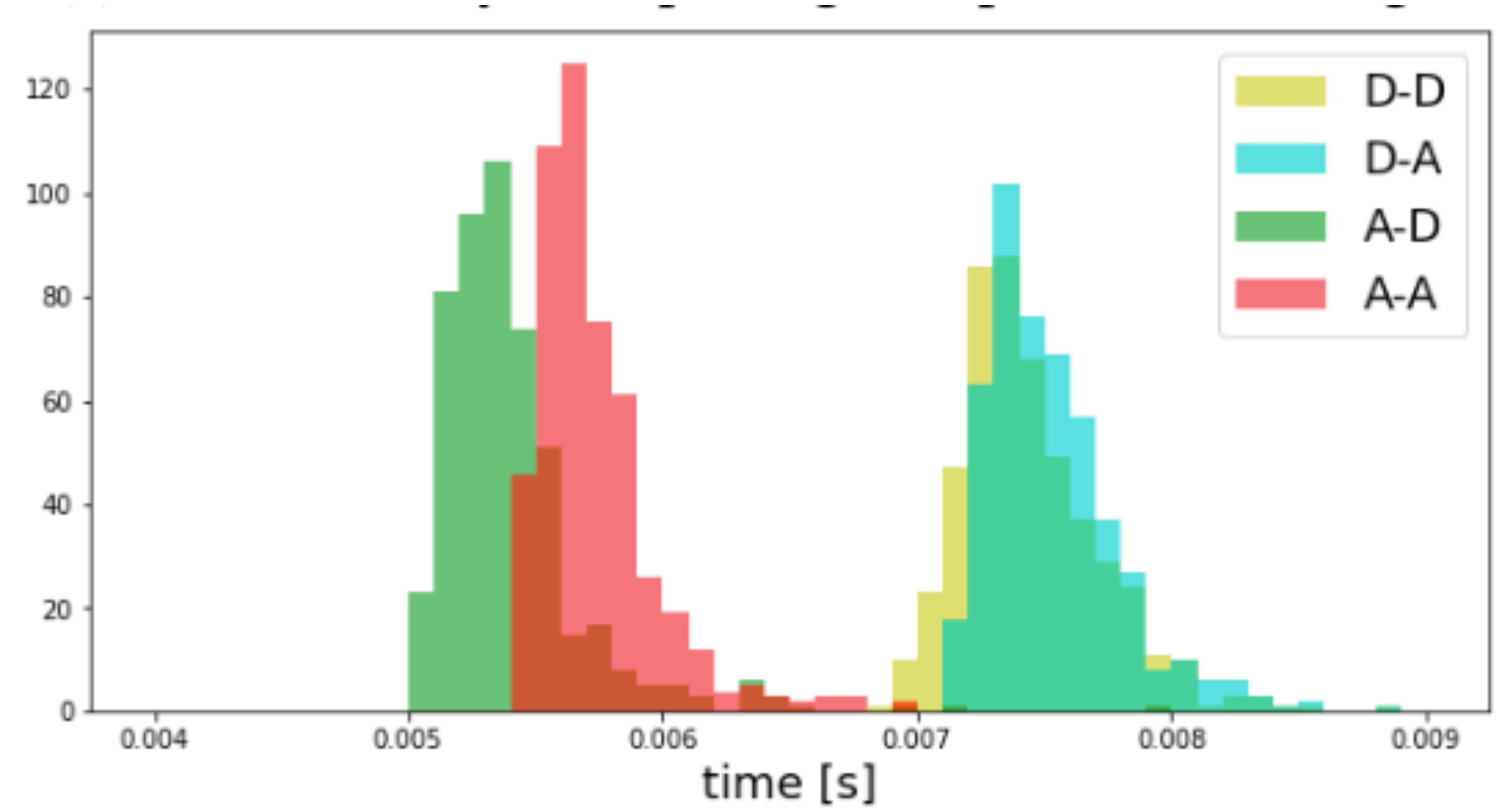
Example : {DD, DA, AD, AA}



Performance Comparison for Scientific Computations on the Edge via Relative Performance
Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden
PAISE Workshop | May 21, 2021

Clustering Methodology

Example : {DD, DA, AD, AA}



Steps:

1) Measure each algorithm N times.
2) Compare adjacent pairs of algorithms and sort them.
3) Repeat step (2) K times to compute relative scores.

Algorithm	DD	AA	DA	AD
Rank	1	2	3	4
	"DD" < "AA"			

Algorithm	AA	DD	DA	AD
Rank	1	2	3	4
		"DD" ~ "DA"		

Algorithm	AA	DD	DA	AD
Rank	1	2	2	3
			"DA" < "AD"	

Algorithm	AA	DD	AD	DA
Rank	1	2	2	2
		"DD" < "AD"		

Algorithm	AA	AD	DD	DA
Rank	1	2	3	3
	"AA" < "AD"			

Algorithm	AD	AA	DD	DA
Final Rank	1	2	3	3

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Modern Inverse Problems

RWTH AACHEN UNIVERSITY

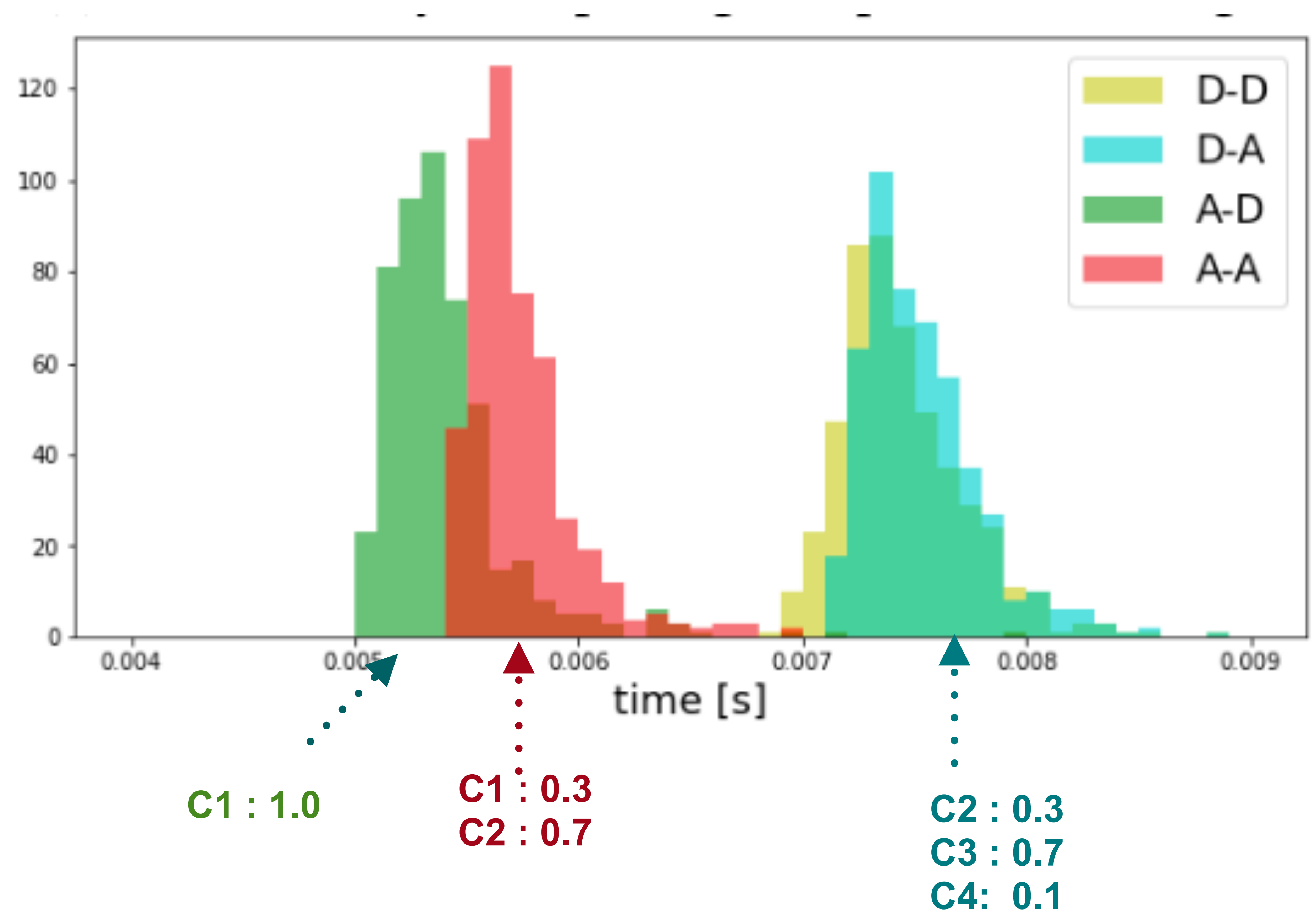
Clustering Methodology

Steps:

1) Measure each algorithm N times.
2) Compare adjacent pairs of algorithms and sort them.
3) Repeat step (2) K times to compute relative scores.

Cluster	*Algorithm*	*Relative Score*
C1	AD AA	1.0 0.3
C2	AA DD DA	0.7 0.3 0.3
C3	DD DA	0.7 0.6
C4	DA	0.1

Example : {DD, DA, AD, AA}



Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

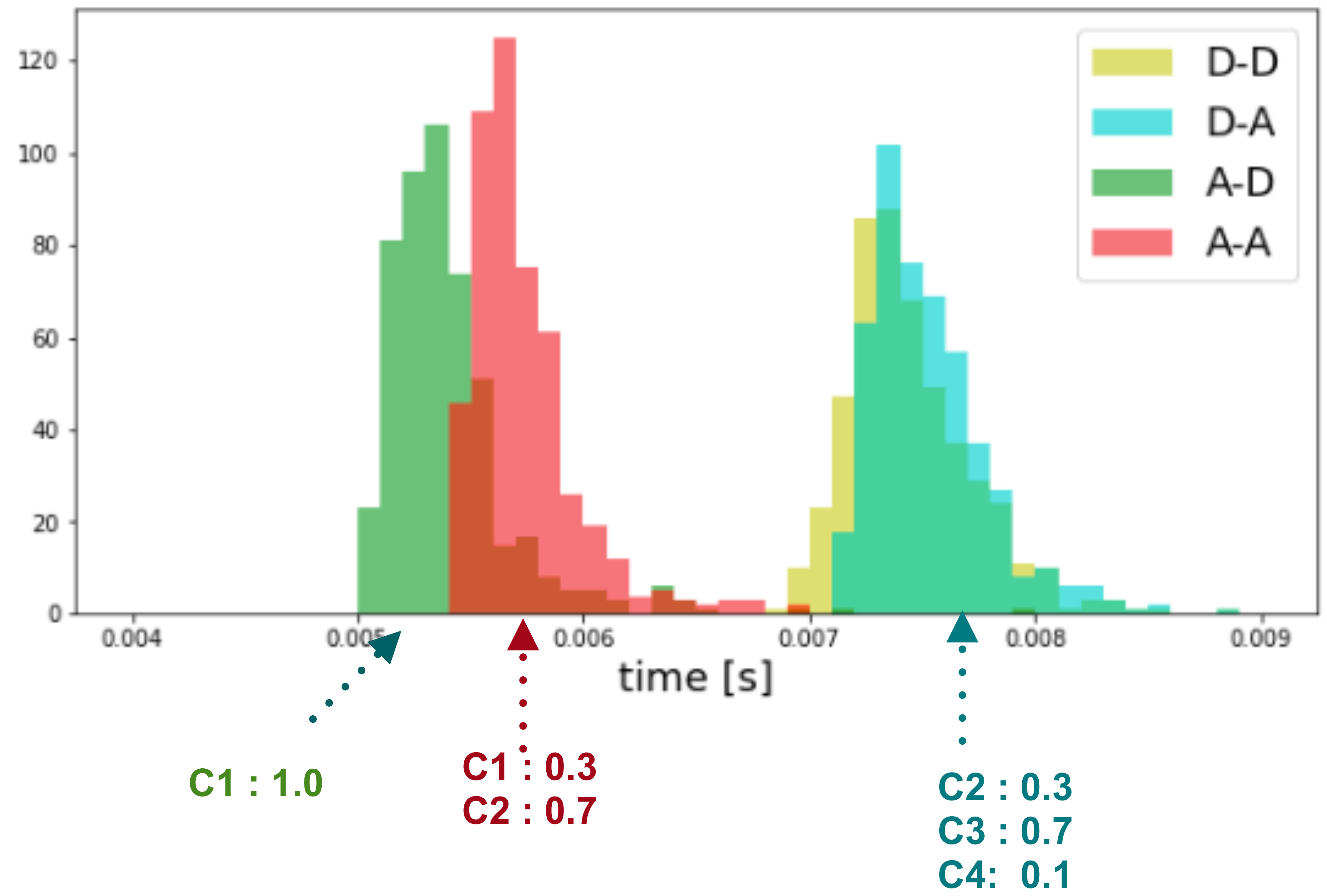
Clustering Methodology

Steps:

1) Measure each algorithm N times.
2) Compare adjacent pairs of algorithms and sort them.
3) Repeat step (2) K times to compute relative scores.

Cluster	*Algorithm*	*Relative Score*
C1	AD AA	1.0 0.3
C2	AA DD DA	0.7 0.3 0.3
C3	DD DA	0.7 0.6
C4	DA	0.1

Example : {DD, DA, AD, AA}



Each cluster is NOT a SET, but a sequence.
The order or algorithms within a cluster is important!

Summary

- We consider scientific codes that can have many different ways of implementation.
- Cluster the implementations into performance classes.
- Our clustering methodology creates a discrimination among the candidate implementations.
- The information of discrimination can be used as a basis to train neural network based approaches.

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden

Summary

- We consider scientific codes that can have many different ways of implementation.
- Cluster the implementations into performance classes.
- Our clustering methodology creates a discrimination among the candidate implementations.
- The information of discrimination can be used as a basis to train neural network based approaches.

Acknowledgement

Financial support from the
Deutsche Forschungsgemeinschaft (DFG)
through grant IRTG-2379 is
gratefully acknowledged

Aravind Sankaran (aravind.sankaran@rwth-aachen.de) | RWTH Aachen University, Germany
Paolo Bientinesi (pauldj@cs.umu.se) | Umeå University, Sweden