Scaling Python Data Applications with Dask

INTRODUCTION

Paweł Kordek
SOFTWARE ENGINEER

@pawel_kordek https://kordek.github.io

Dask

Course overview

Understanding Dask

User interfaces

Target Audience

Target Audience

Any level of Python and data processing experience.

Target Audience

Any level of Python and data processing experience.

Looking for ways to easily accommodate ‘bigger’ data.

Python for Data

Rich ecosystem

Python for Data

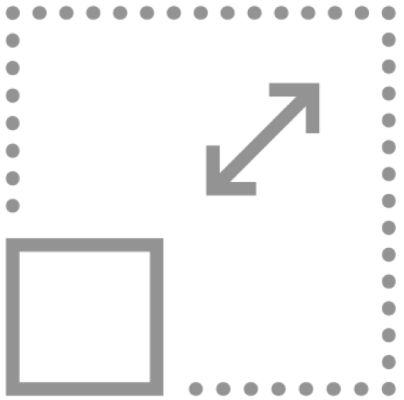
Rich ecosystem

Accessible

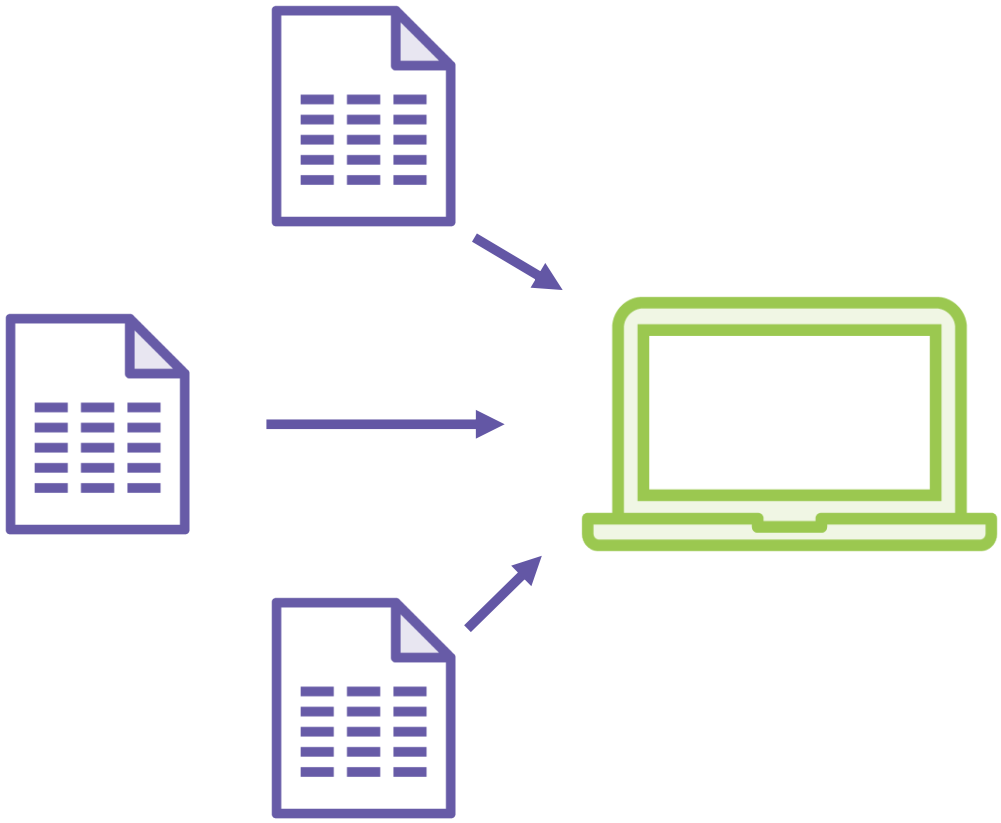
Python for Data

Rich ecosystem

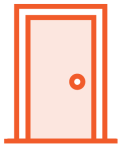
Accessible



Scaling is problematic



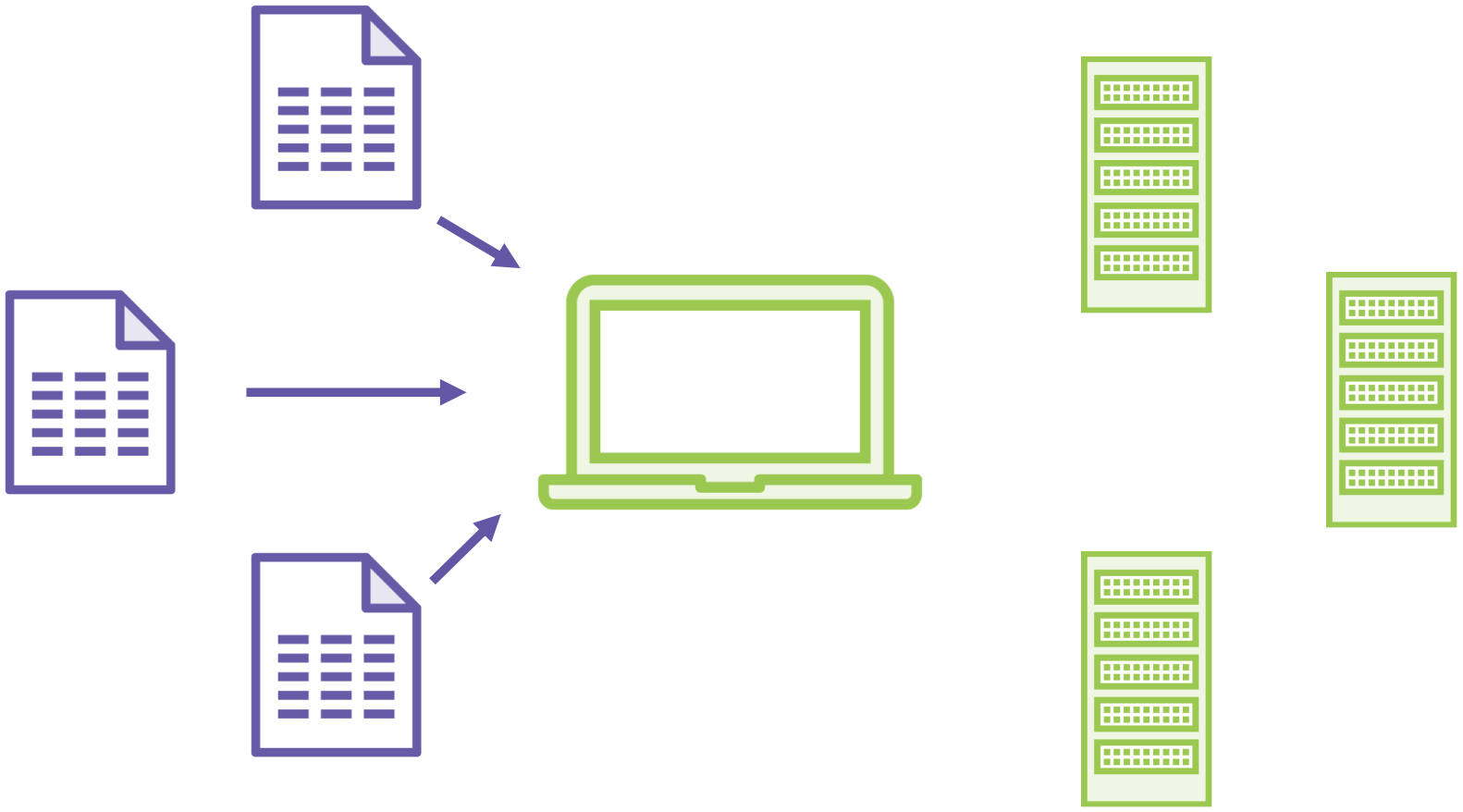
Change Technology?



Leaving familiar environment

Need for additional tailoring

Operational complexity



Few Changes to Your Python Code

Course Components

- Understanding Dask
- Basic Usage
- Internals and Monitoring
- Numpy and Pandas
- Multiple Machines

Key Outcome

Confidence in writing Python data applications that scale.

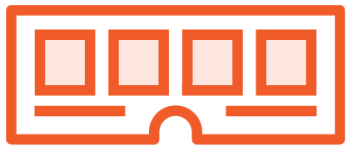
Understanding Dask

Scalability

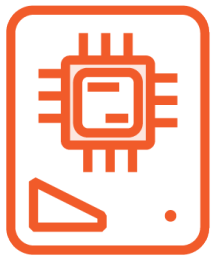
Process larger datasets

Perform more intensive computations

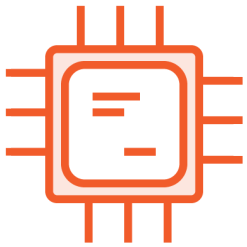
By utilizing additional resources



Memory

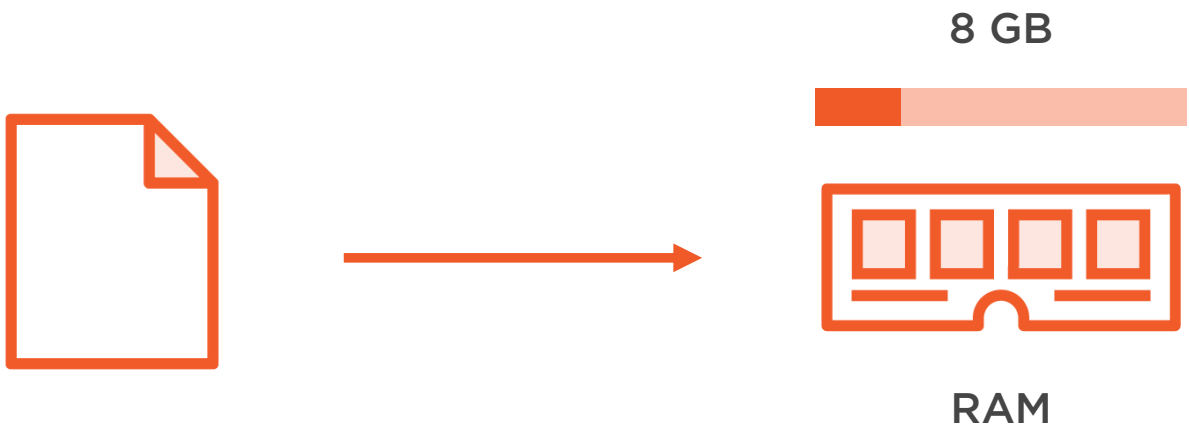


Disk

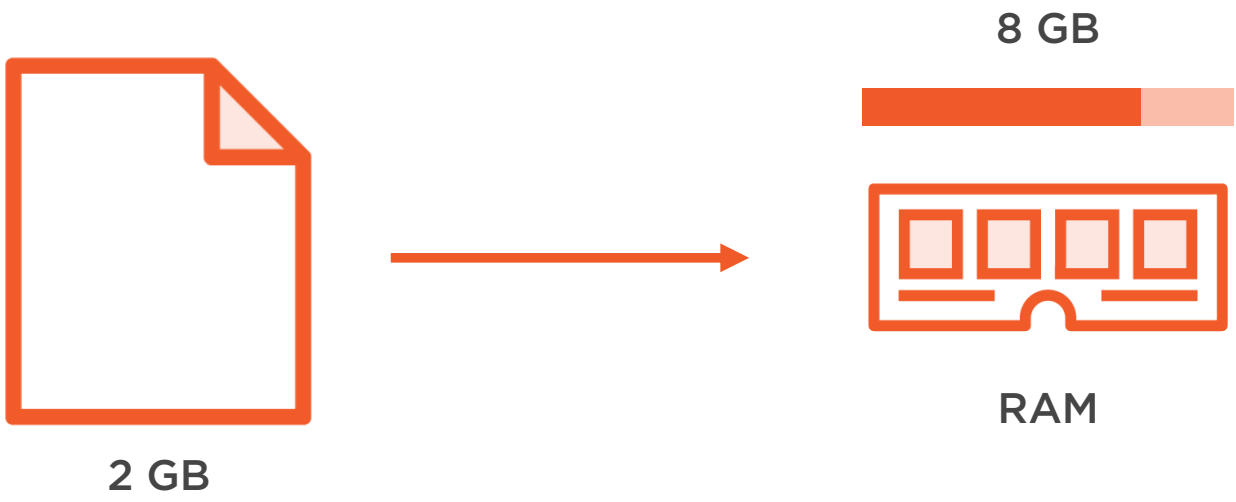


Processor

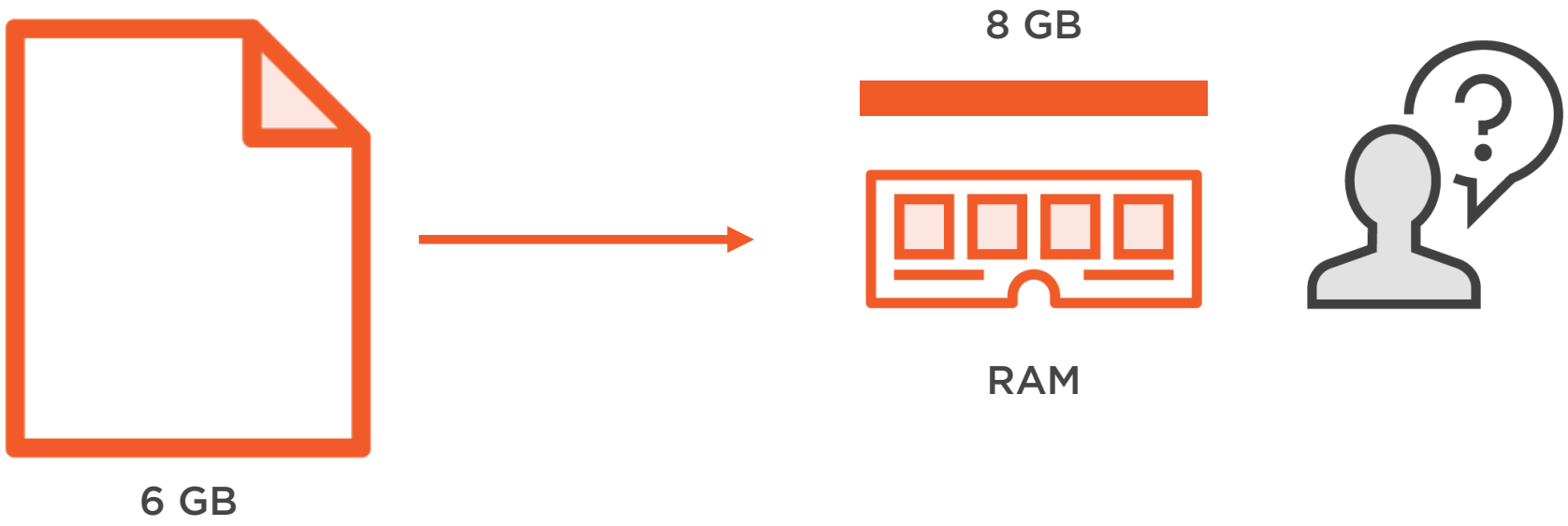
Dataset Size Limit



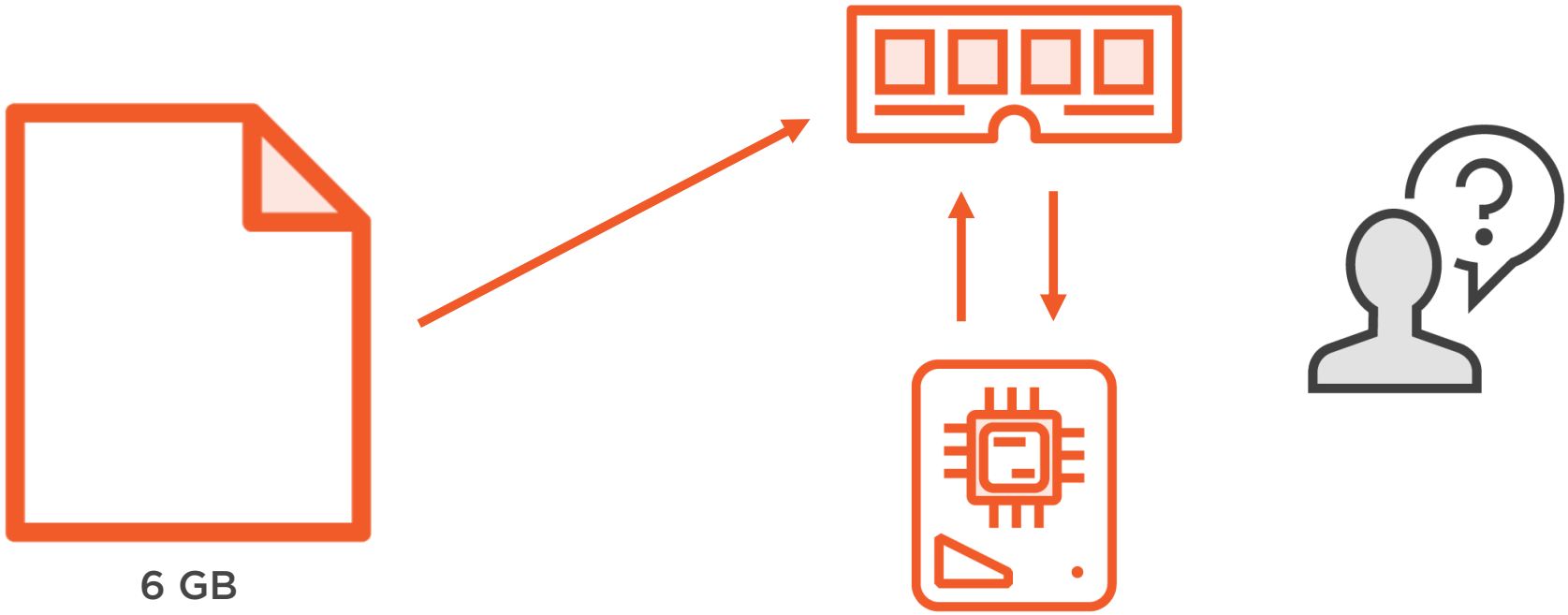
Dataset Size Limit



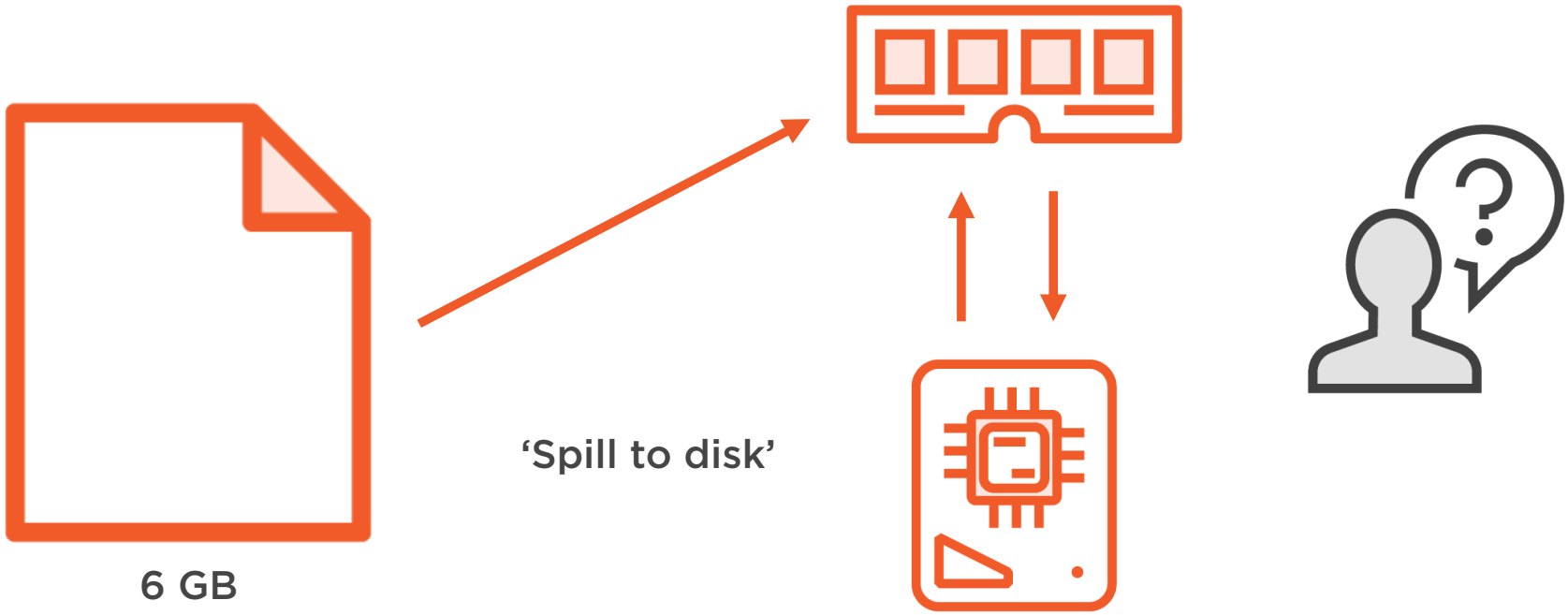
Dataset Size Limit

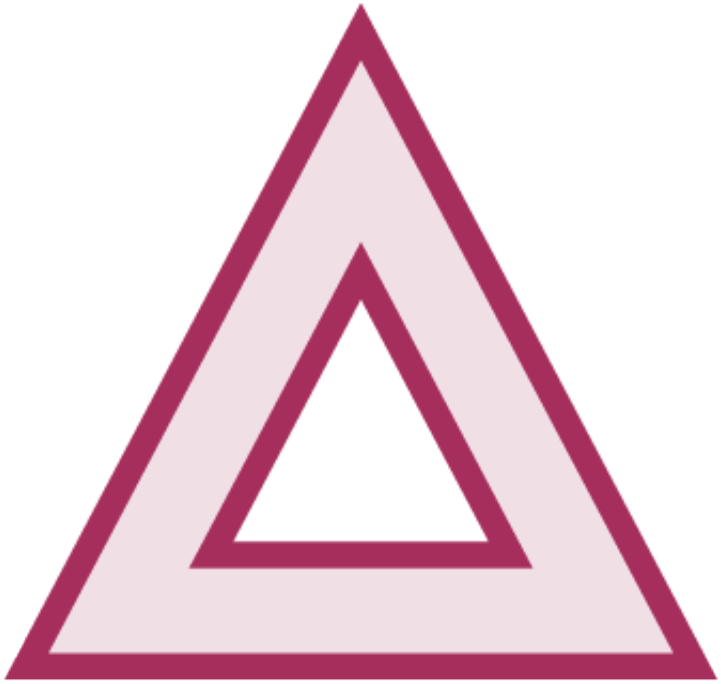


Dataset Size Limit



Dataset Size Limit

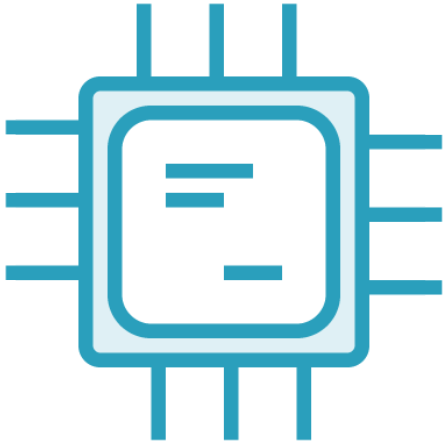




This is not straightforward

Limited Parallelism

Limited Parallelism



Limited Parallelism

	Count
A	0
B	0

Limited Parallelism

	Count
A	1
B	0

Limited Parallelism

	Count
A	1
B	1

Limited Parallelism

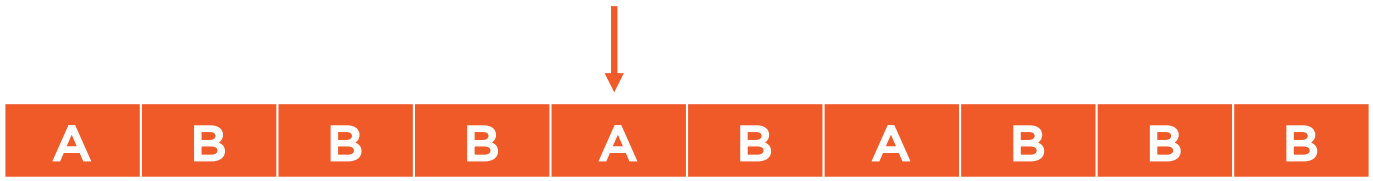
	Count
A	1
B	2

Limited Parallelism

	Count
A	1
B	3

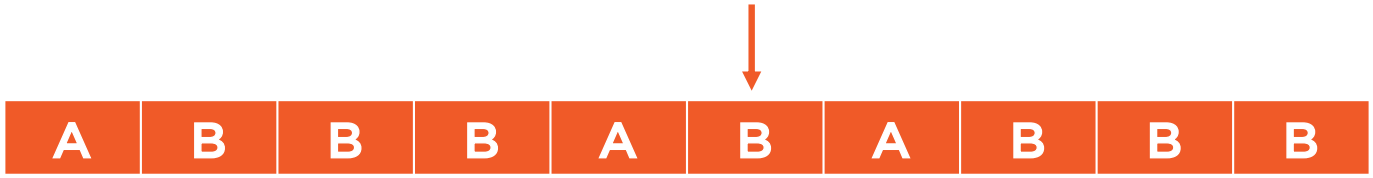
Limited Parallelism

	Count
A	2
B	3



Limited Parallelism

	Count
A	2
B	4



Limited Parallelism

	Count
A	3
B	4

Limited Parallelism

	Count
A	3
B	5

Limited Parallelism

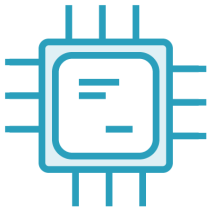
	Count
A	3
B	6

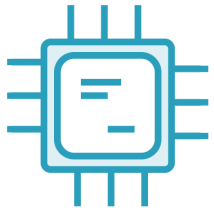
Limited Parallelism

	Count
A	3
B	7

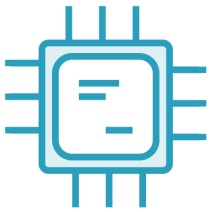
	Count
A	0
B	0

	Count
A	0
B	0

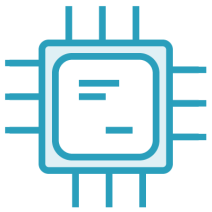




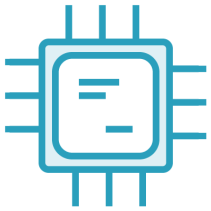
	Count
A	1
B	0



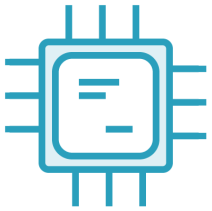
	Count
A	0
B	1



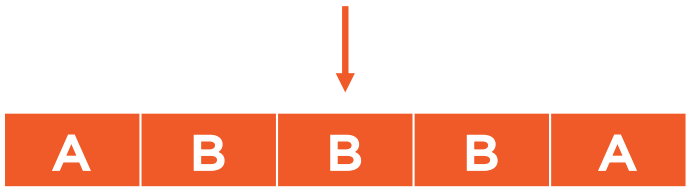
	Count
A	1
B	1

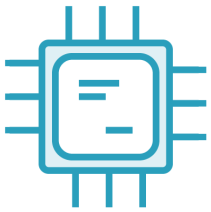


	Count
A	1
B	1

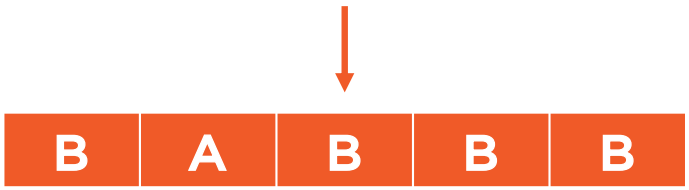


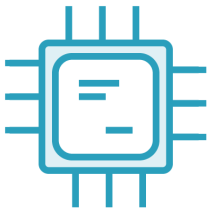
	Count
A	1
B	2



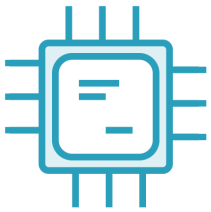


	Count
A	1
B	2

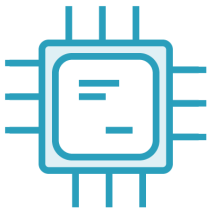




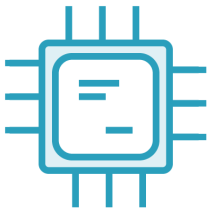
	Count
A	1
B	3



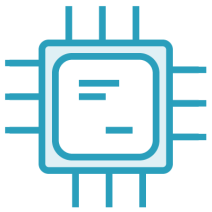
	Count
A	1
B	3



	Count
A	2
B	3



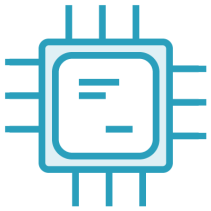
	Count
A	1
B	4

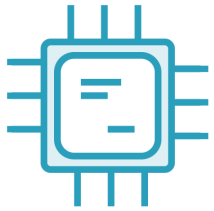


	Count
A	2
B	3

	Count
A	3
B	7

	Count
A	1
B	4





Challenges

No magic here

Explicit instructions needed

Parallel Computing and Python

Threads

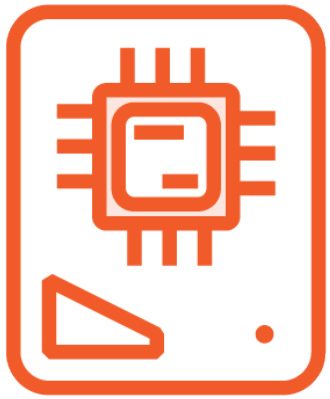
GIL

Processes

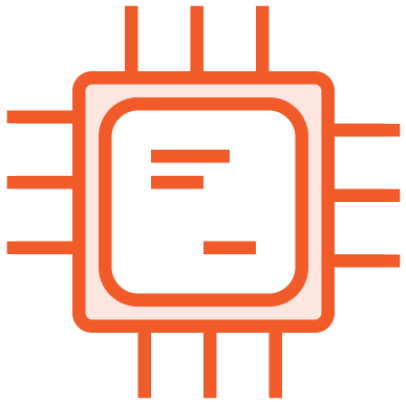
Cython/C

Not Ubiquitous

Underutilization



Hard disk



CPU/Server

APIs

Bag

Unstructured and semi-structured data

```
# Can contain e.g. JSON objects
# { ‘a’: 2, ‘b’: 4 }
```

Bag

Unstructured and semi-structured data

```
# Can contain e.g. JSON objects
# { ‘a’: 2, ‘b’: 4 }

some_bag \
  .map(lambda x: x[‘b’]) \
  .distinct()
```

Bag

Unstructured and semi-structured data

Array

Numerical computations – subset of NumPy

```
# Matrix operations
# [[1, 2],
#  [3, 5]]
```

Array

Numerical computations – subset of NumPy

```
# Matrix operations
# [[1, 2],
#  [3, 5]]

a1 * a2  # Matrix multiplication

a1 + a2  # Row summation
```

Array

Numerical computations – subset of NumPy

DataFrame

Replicates subset of Pandas API

```
#   c d
# a 1 3
# b 2 3

df.loc[df.loc[‘c’] == 1, ‘d’]  # Gives 3
```

DataFrame

Replicates subset of Pandas API

```
#   c d
# a 1 3
# b 2 3

df.loc[df.loc[‘c’] == 1, ‘d’]  # Gives 3

df.merge(other_df)

df.groupby(‘d’).sum()
```

DataFrame

Replicates subset of Pandas API

Lower-level APIs

dask.delayed

dask.futures

Other Tools

Spark	Dask
SQL engine	
Streaming engine	
JVM	

Other Tools

Spark	Dask
SQL engine	More lightweight
Streaming engine	Flexible (at some performance cost)
JVM	Accessible

Other Tools

Spark	Dask
SQL engine	More lightweight
Streaming engine	Flexible (at some performance cost)
JVM	Accessible
Standalone	

Other Tools

Spark	Dask
SQL engine	More lightweight
Streaming engine	Flexible (at some performance cost)
JVM	Accessible
Standalone	Integrates with other libraries

Let’s go!