

Parallel Programming in Python – a very, very short introduction

Agenda

1. Computer architectures, data structures, and opportunities
2. Case 1 – computing a distance matrix
3. Case 2 – computing the average of a large sample

The scope of the technique that will be presented:

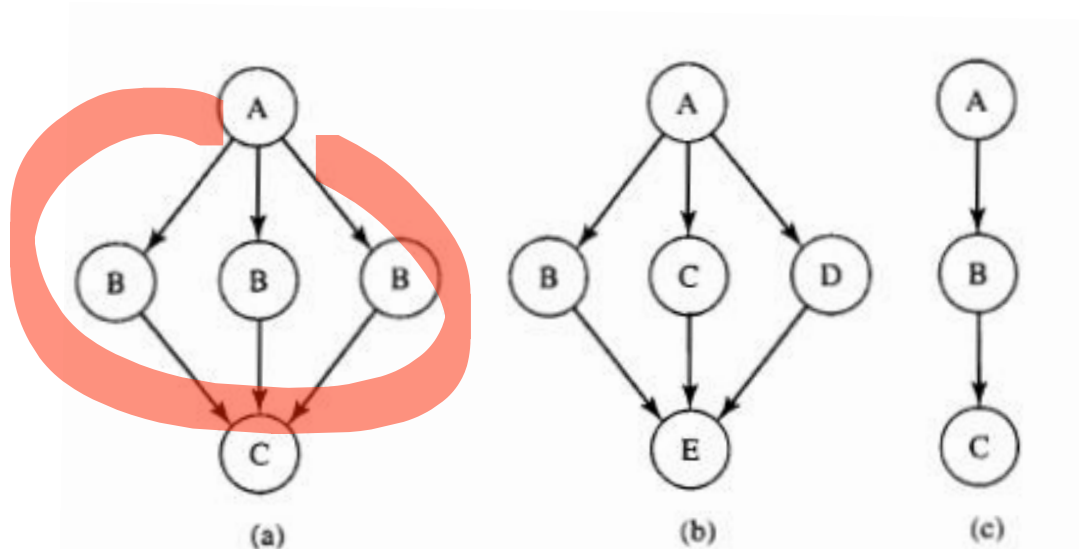


Figure 1.4 Parallelism in data dependence graphs. Vertices represent tasks. The letter inside a vertex indicates the operation being performed. Edges denote dependences among tasks. (a) A graph exhibiting data parallelism. Three tasks may concurrently apply operation B to different operands. (b) A graph exhibiting functional parallelism. Tasks performing operations B, C, and D may be performed concurrently. (c) A purely sequential dependence graph. However, if all tasks take the same amount of time to execute and multiple problem instances need to be processed, operation C may be performed on instance i while operation B is performed on instance $i + 1$ and operation A is performed on instance $i + 2$. This structure is called pipelining.

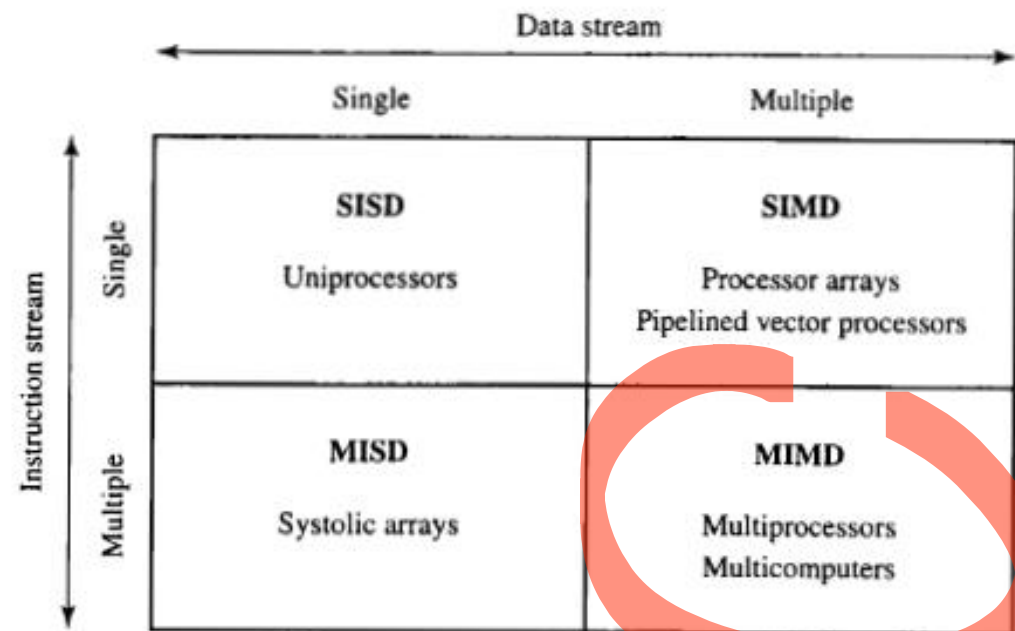


Figure 2.20 Flynn's taxonomy of computer architectures.

Resumo. *Aplicações Bag-of-Tasks (BoT) são aplicações paralelas compostas de tarefas independentes (ou seja, embaraçosamente paralelas), que não se comunicam entre si, podem depender de um ou mais arquivos de entrada e podem ser executadas em qualquer ordem.*



Case 1 – computing a distance matrix

Case 1 - computing a distance matrix, sequential scheme

k-dimensional
vectors

v1	v2	v3		vk
----	----	----	--	----



distance
matrix

	v1	v2	v3		vk
v1	0				
v2		0			v
v3			0		
				0	
vk		v			0

tasks

v1	v1	v1	v1	v2	v2	v2	v3	v3	
v2	v3		vk	v3		vk		vk	



Sequential execution, process P1



--	--	--	--	--	--	--	--	--	--

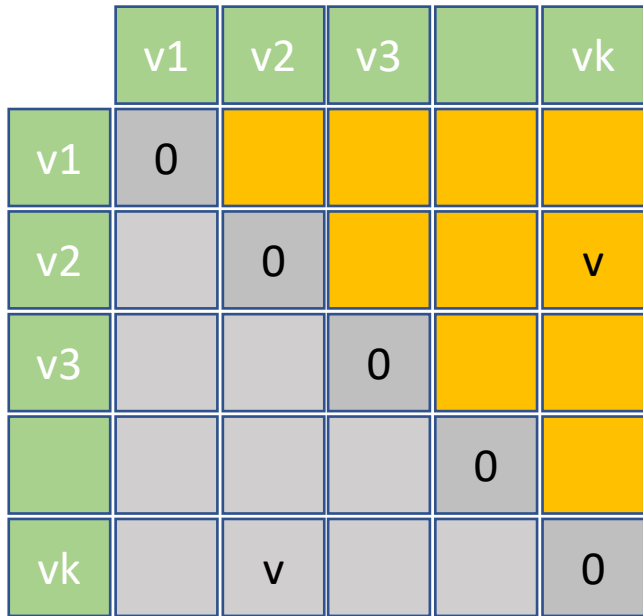
distance matrix

Case 1 - computing a distance matrix, parallel scheme

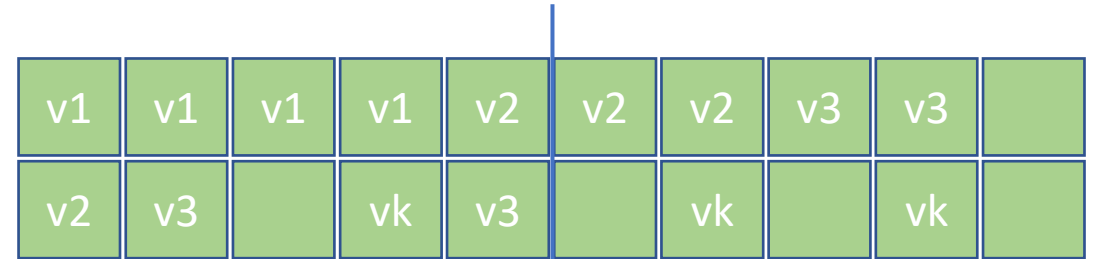
k-dimensional
vectors



distance
matrix



tasks



Parallel execution process P1

Parallel execution process P2

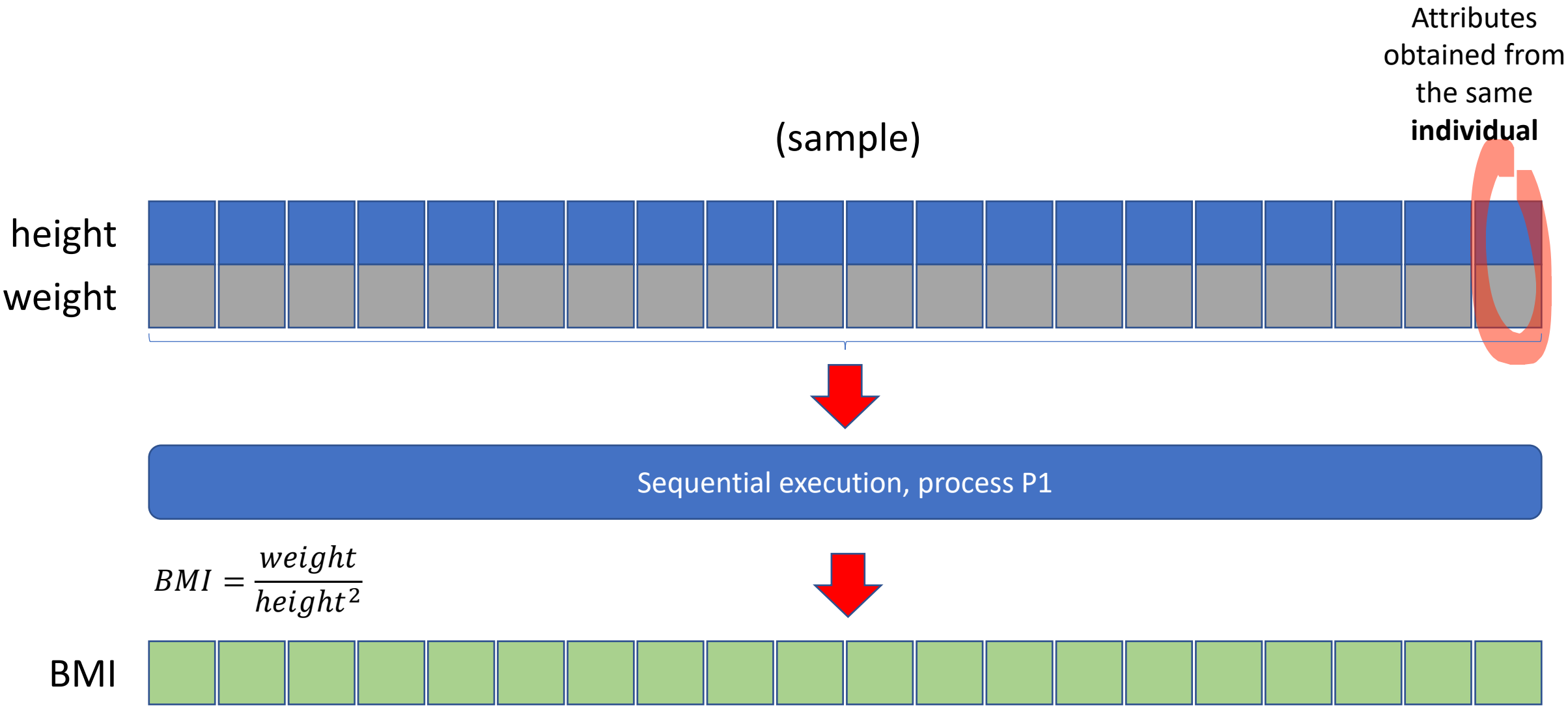


distance matrix

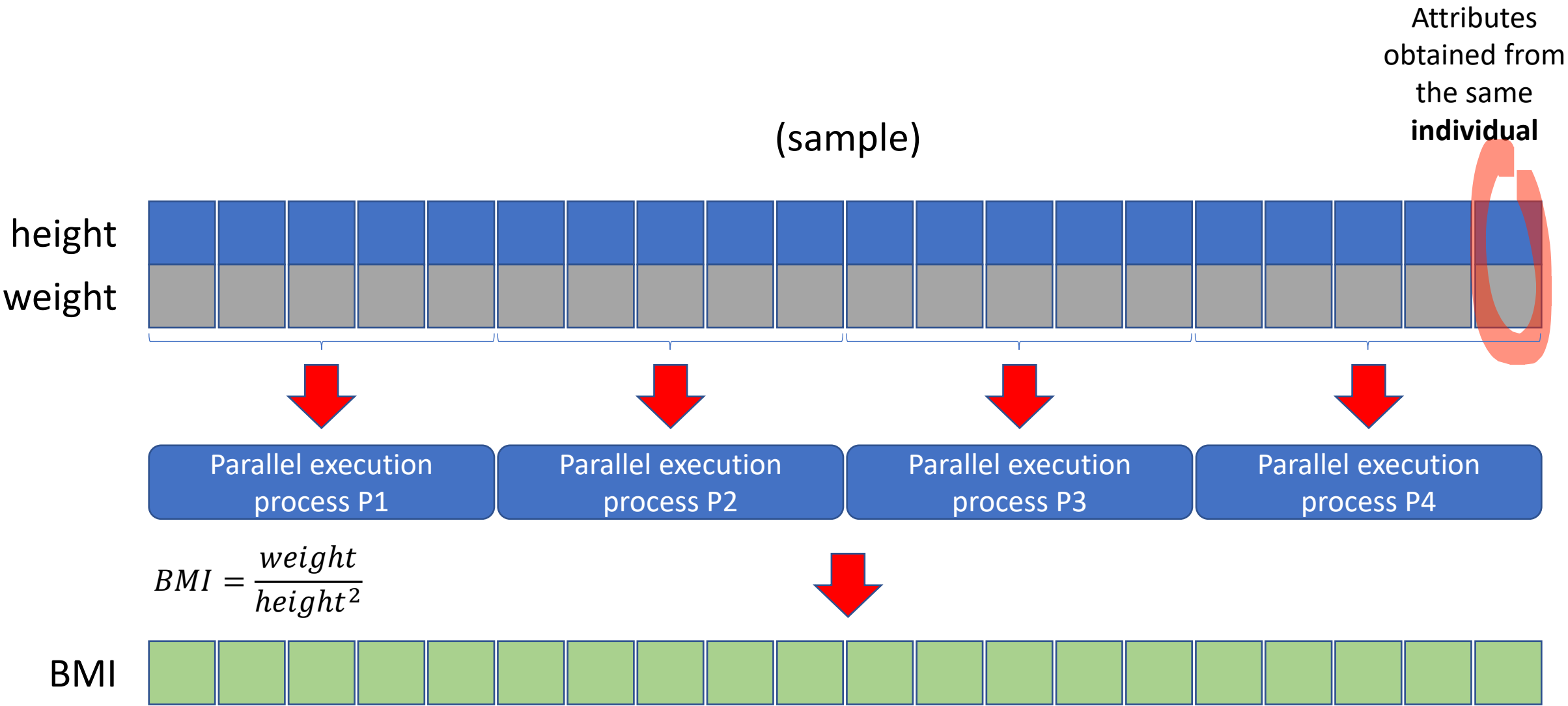


Case 2 – computing the average of a large sample

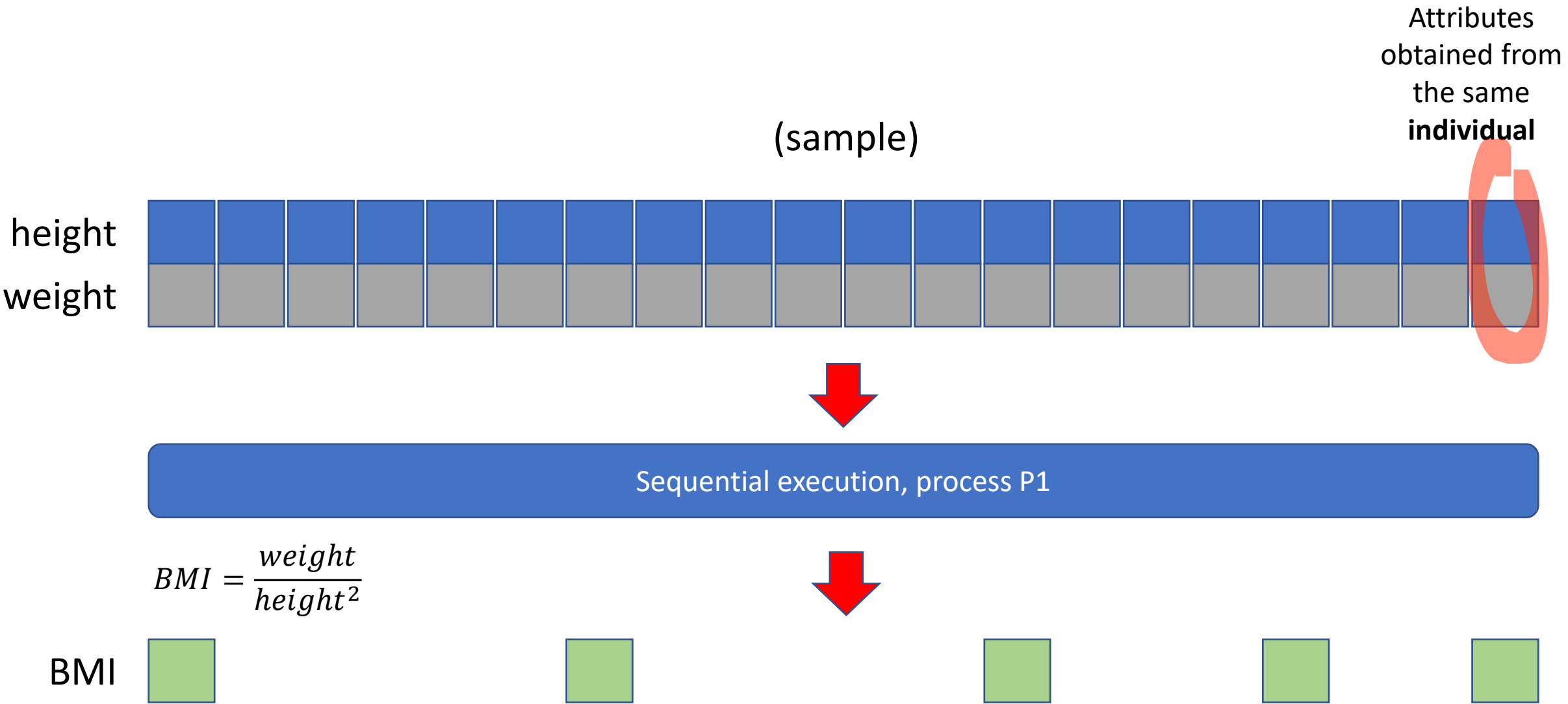
Case 2 - computing the average of a large sample

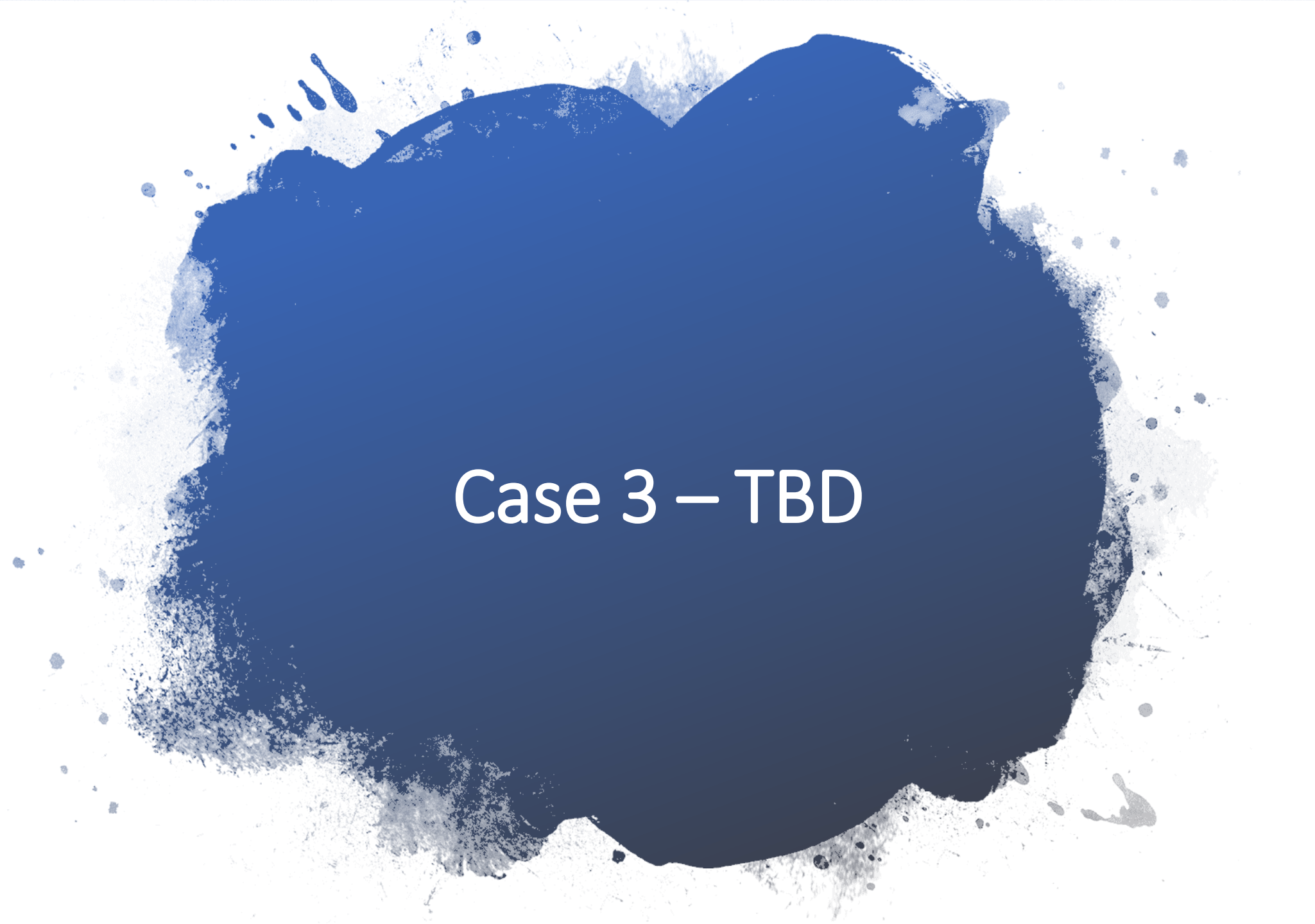


Case 2 - computing the average of a large sample



Case 2 - computing the average of a large sample





Case 3 – TBD

To be resumed in a future meeting ;)

Parallel Programming in Python – a very, very short introduction

Thanks!
