Accelerate Python

for a Genetic Programming Project

Bocheng Lin

2024.6.5

Time Comparison for Developing a Project: Coding Time and Execution Time

C++

Python


I'm a programmer
Which programming Languages You Use
Chatgpt
知乎 @艾克斯

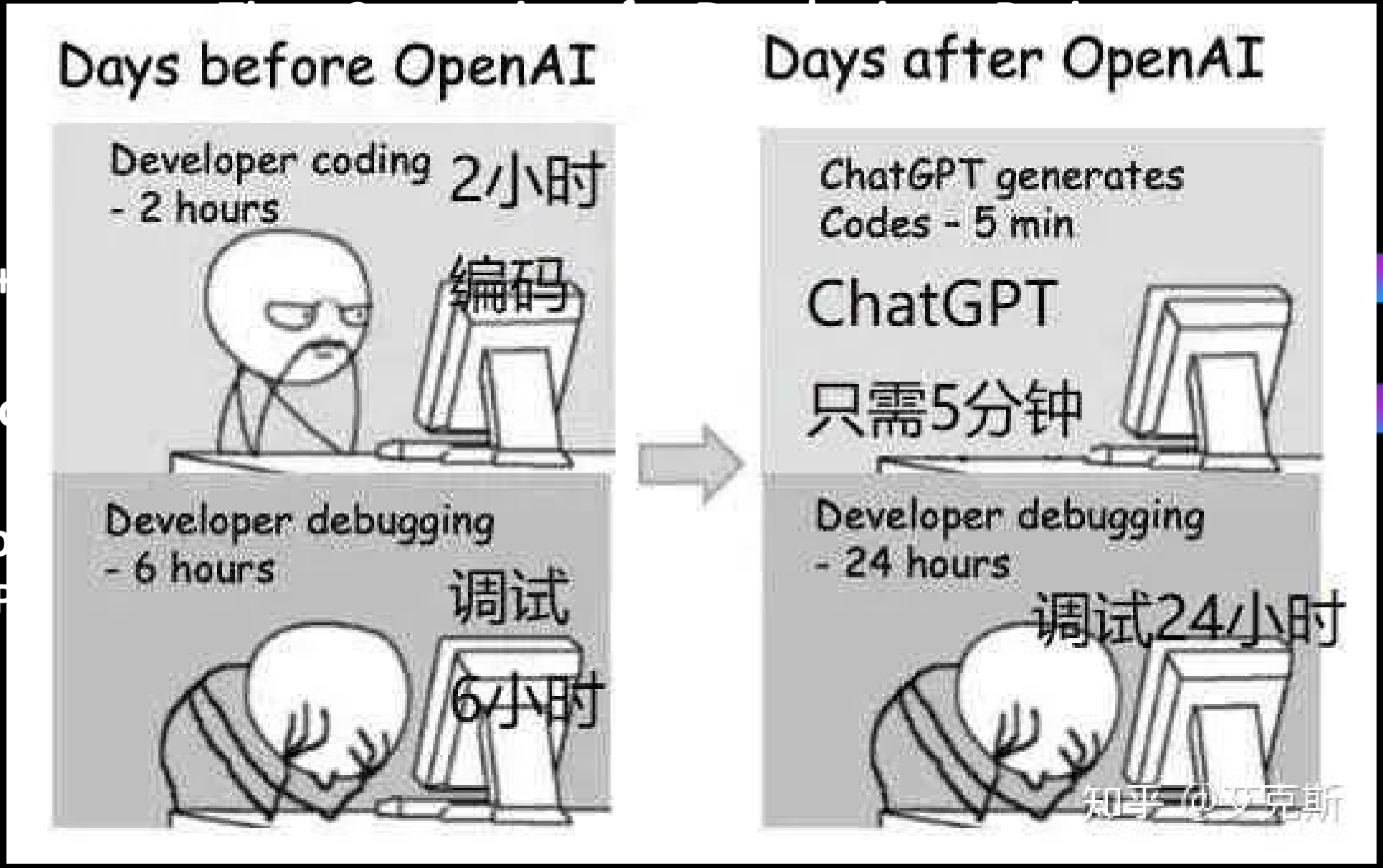
VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

Time Comparison for Developing a Project: Coding Time and Execution Time

C++

Pytho

Pytho
ChatGP



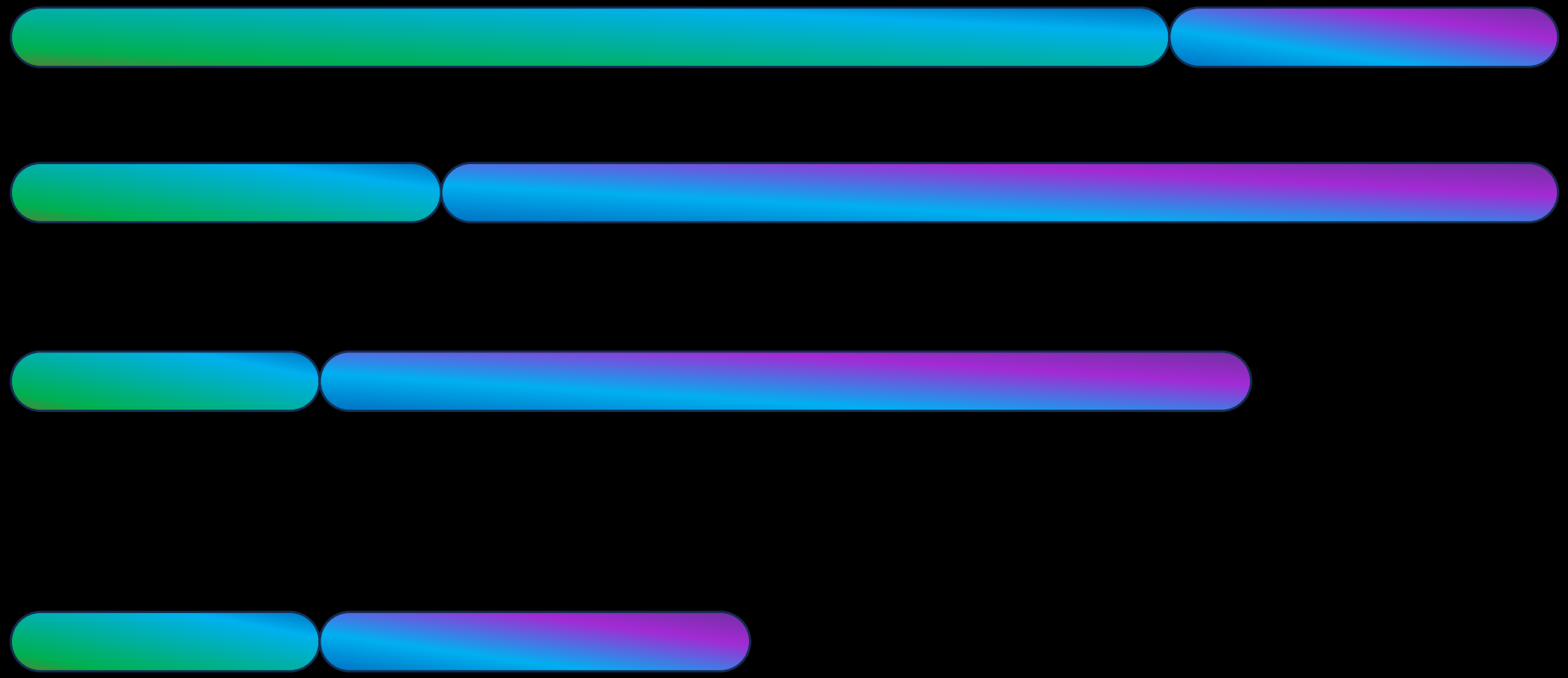
Time Comparison for Developing a Project: Coding Time and Execution Time

C++

Python

Python+ ChatGPT-4

Python+ ChatGPT-4 +?



VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

How to accelerate your python Genetic Programming Codes?

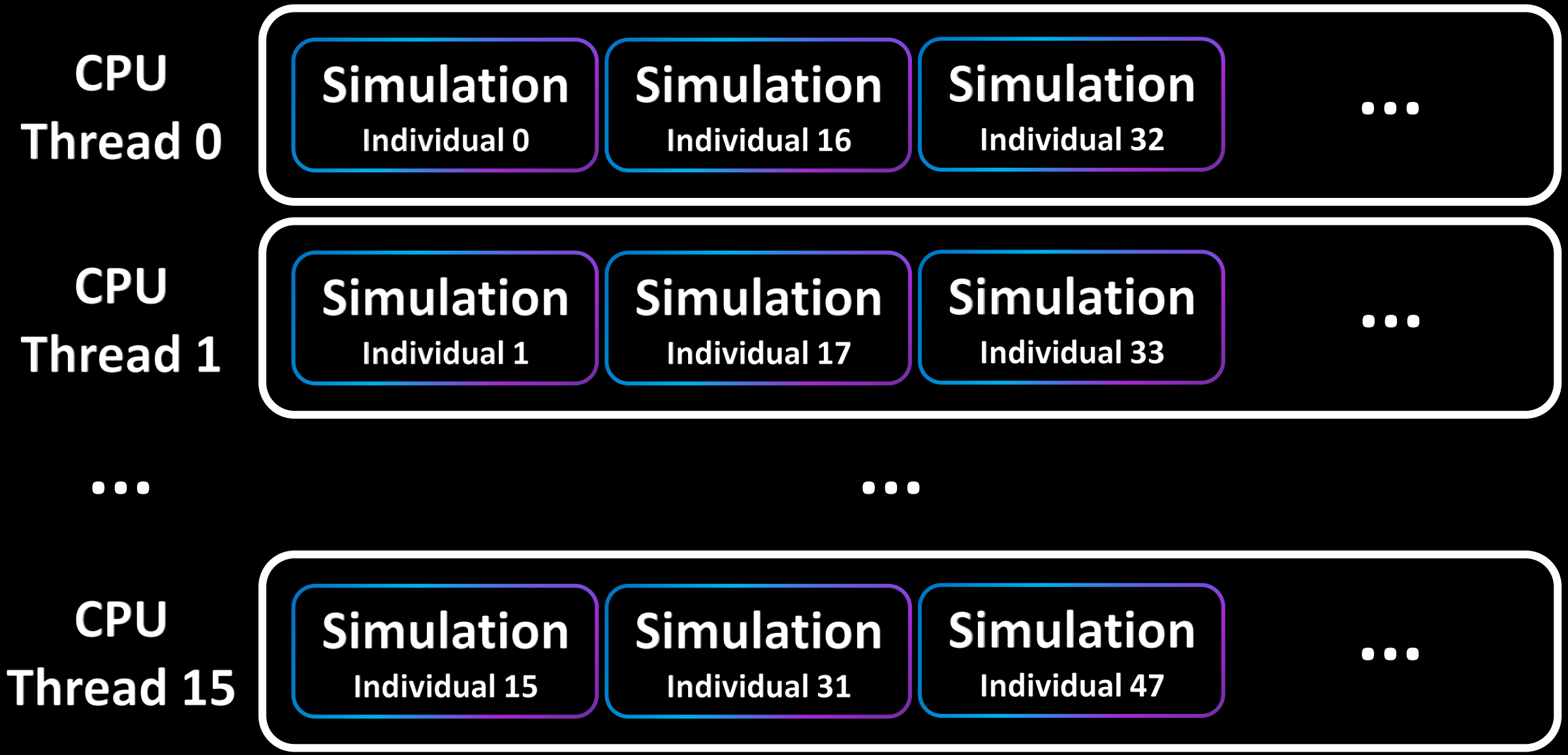
```
for individual in population:
    simulation(individual);
    offspring;
end for
```

```
for individual in population:
        simulation(individual);
        offspring;
end for
```

Parallel

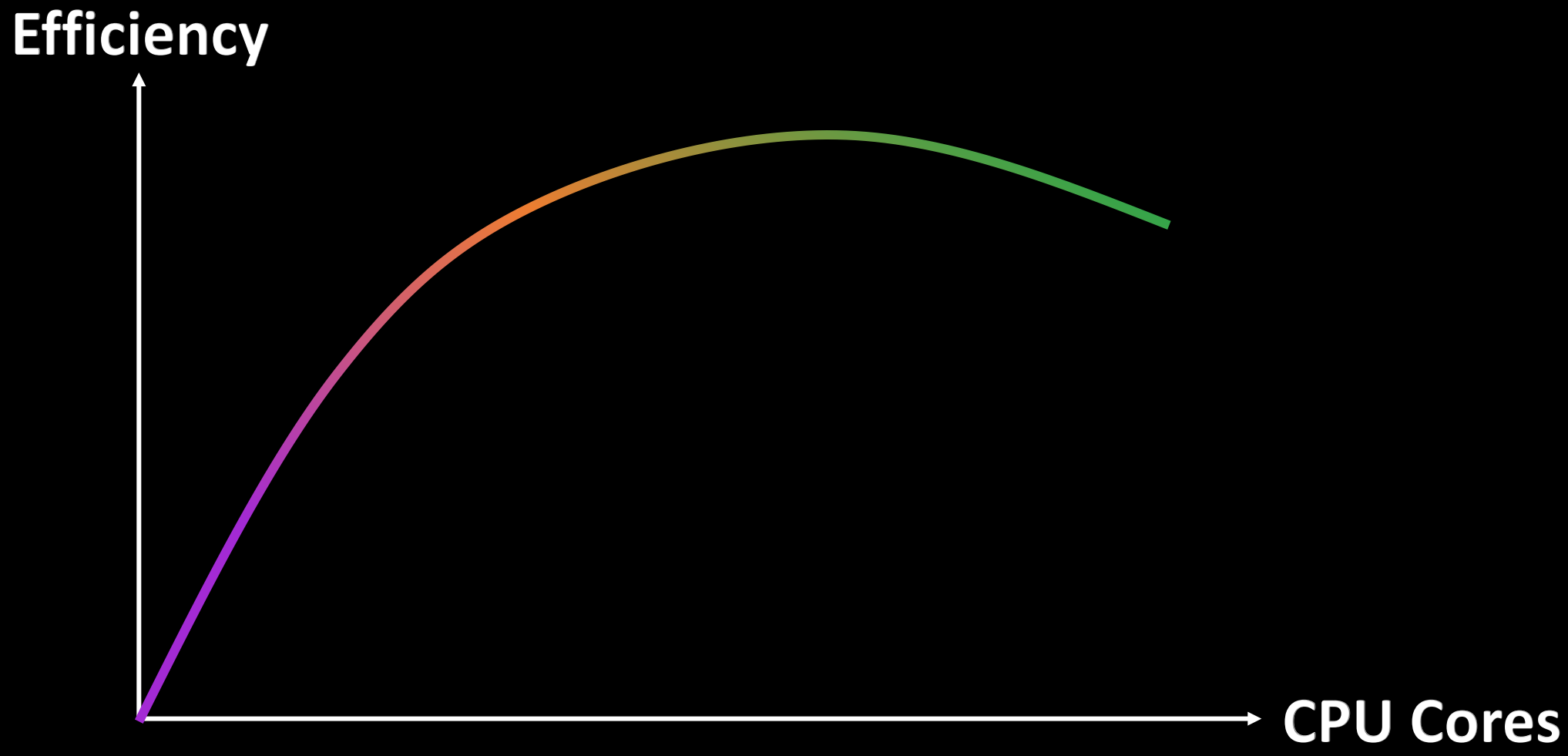
VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

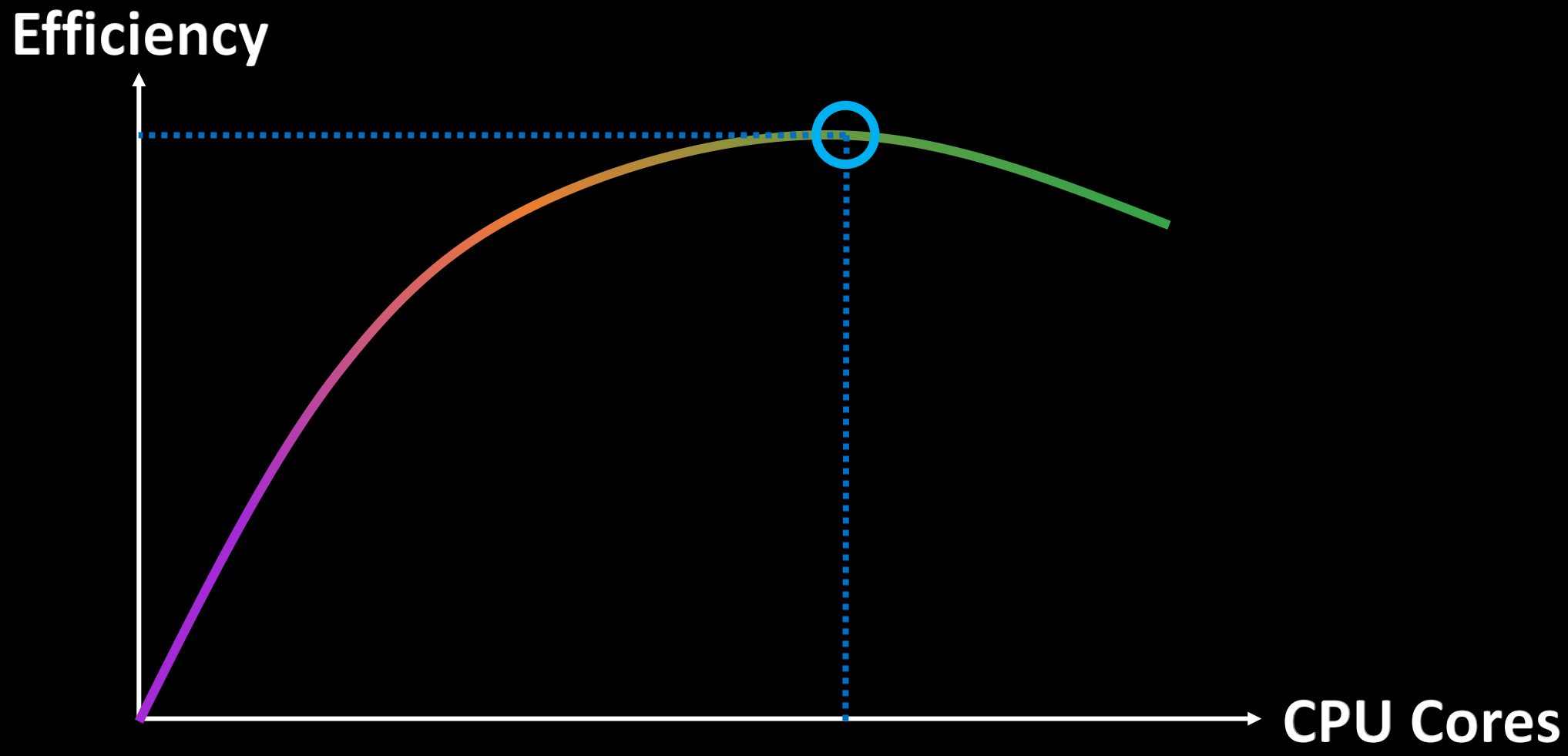
Parallel Simulation/Evaluation

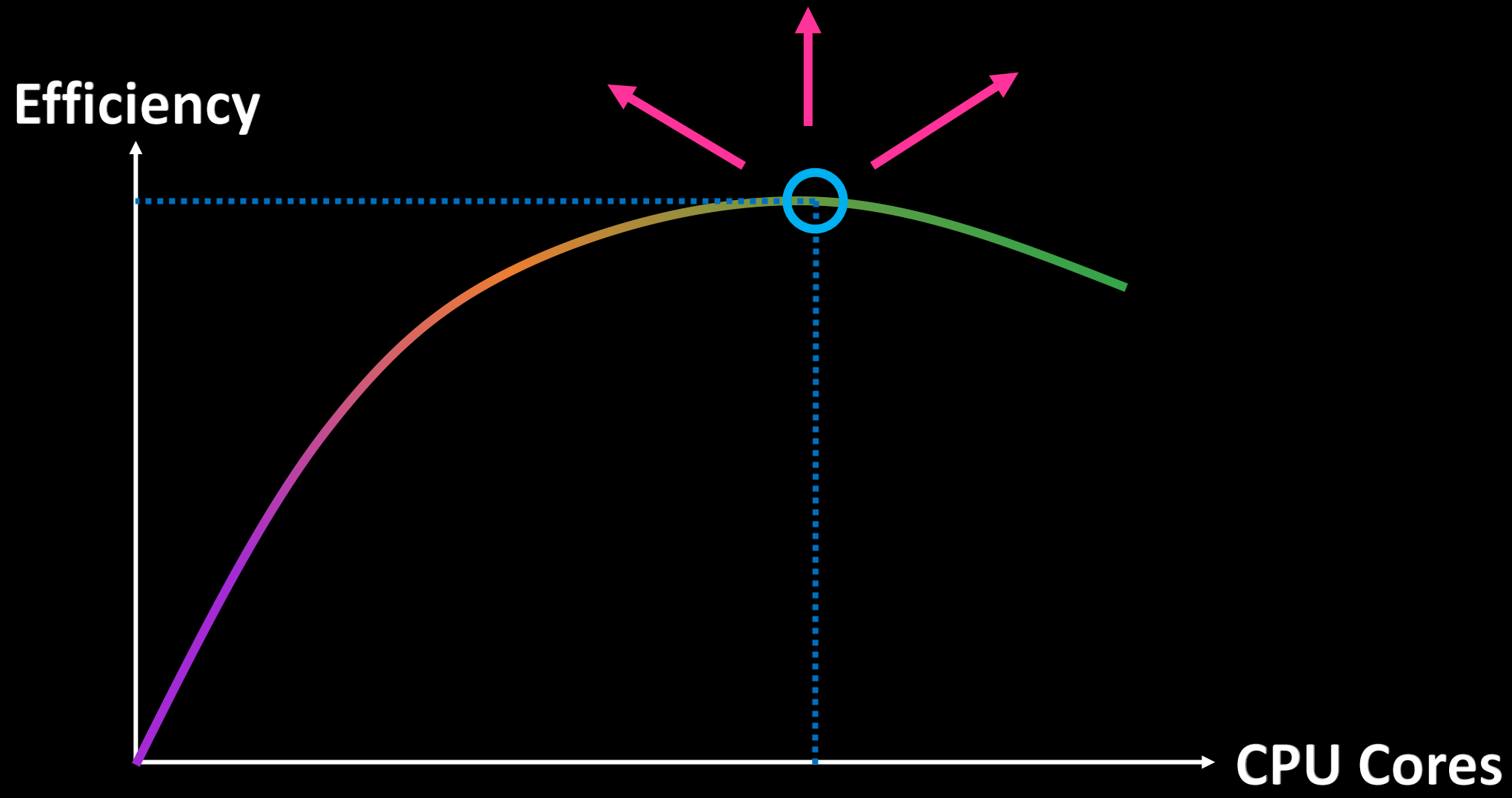


```
# regist
executor = concurrent.futures.ProcessPoolExecutor ( max_workers = cores )
toolbox.register ( "multiprocessing", executor.map )

# parallel evaluation
invalid_ind = [ ind for ind in population ]
fitnesses = toolbox.multiprocessing ( toolbox.evaluate, invalid_ind )
for ind, fit in zip ( invalid_ind, fitnesses ) :
    ind.fitness.values = ( fit, )
```


Efficiency
CPU Cores


Efficiency
CPU Cores


Efficiency
CPU Cores

Go into Simulation

Loops to Matrix

VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

(Matrix A × Matrix B).sum (ms)

Matrix_size	Loops	Numpy	PyTorch_CPU	Numba_Loops
1000	232.2309	1.9934	2.9869	0.9961
2000	817.2810	9.9669	5.9803	1.9991
3000	1793.0334	22.9242	12.9590	1.9944
4000	3219.2883	42.8576	20.9301	4.9837
5000	4919.6362	66.7777	31.8937	6.9737
6000	7170.1484	92.6917	43.8540	10.9632
7000	9691.7591	130.5678	61.7921	15.9428
8000	12531.3115	165.4499	77.7411	17.9372
9000	16748.4260	204.3159	96.6794	21.9247
10000	19626.6966	267.1080	119.6053	27.9043

Intel i7-13700, Python3.8, PyTorch1.7

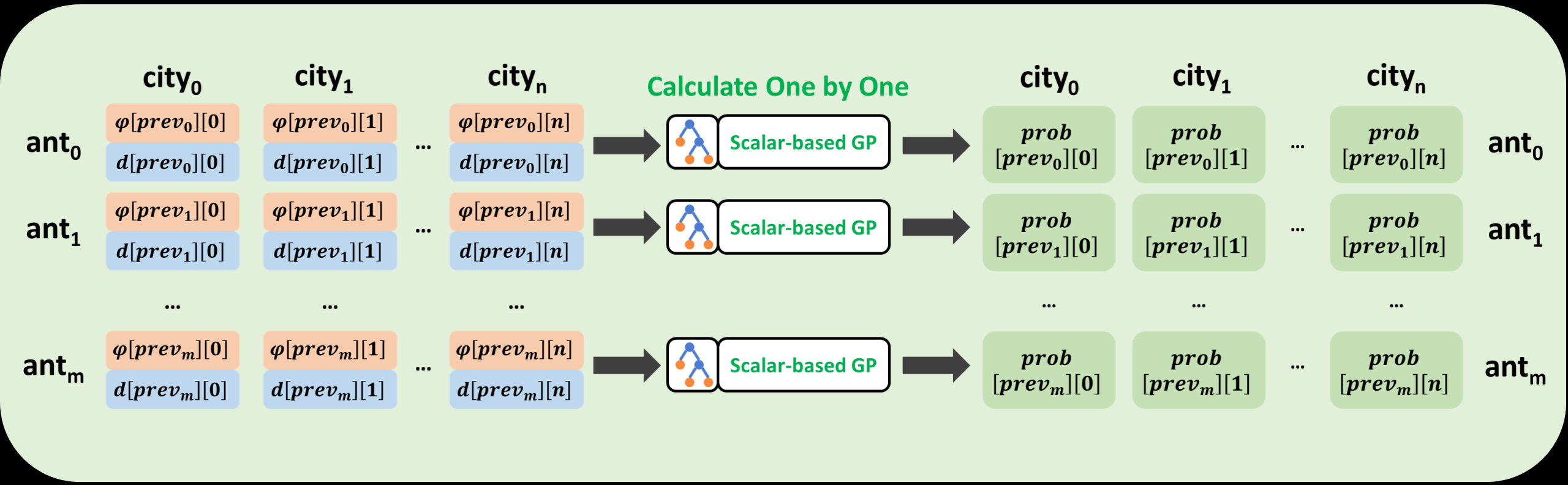
```
for individual in population:
        simulation(individual);
        offspring;
end for
```

```
for individual in population:
    #ACO simulation
    for ant in all_ants:
        while unvisited_cities is not empty:
            for cities in unvisted_cities:
                calculate_priority by individual;
            end for;
            choose next_city;
        end while;
    end for;
    offspring;
end for
```

```
for individual in population:
    #ACO simulation
    for ant in all_ants:
        while unvisited_cities is not empty:
            for cities in unvisted_cities:
                calculate_priority by individual;
            end for;
            choose next_city;
        end while;
    end for;
    offspring;
end for
```

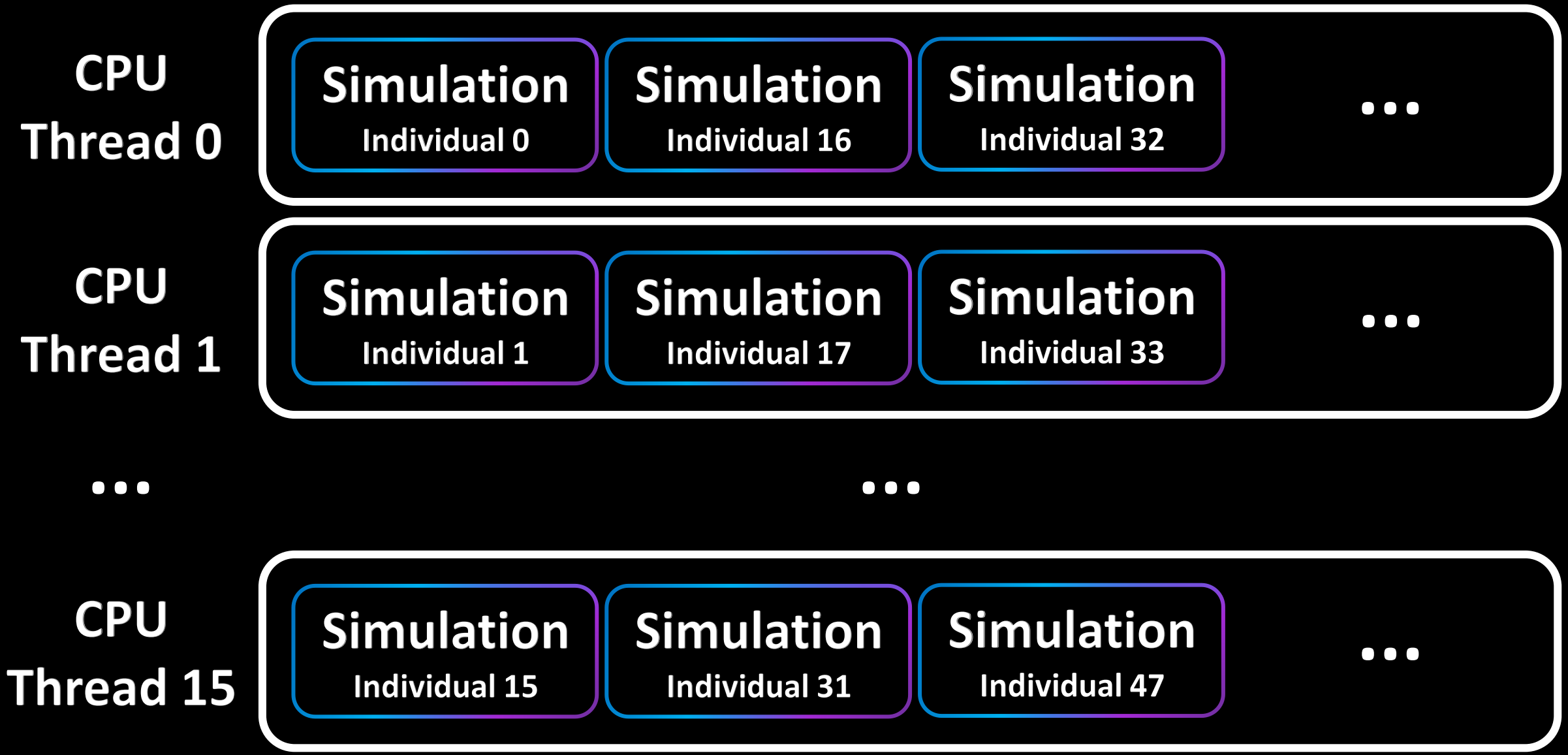
VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

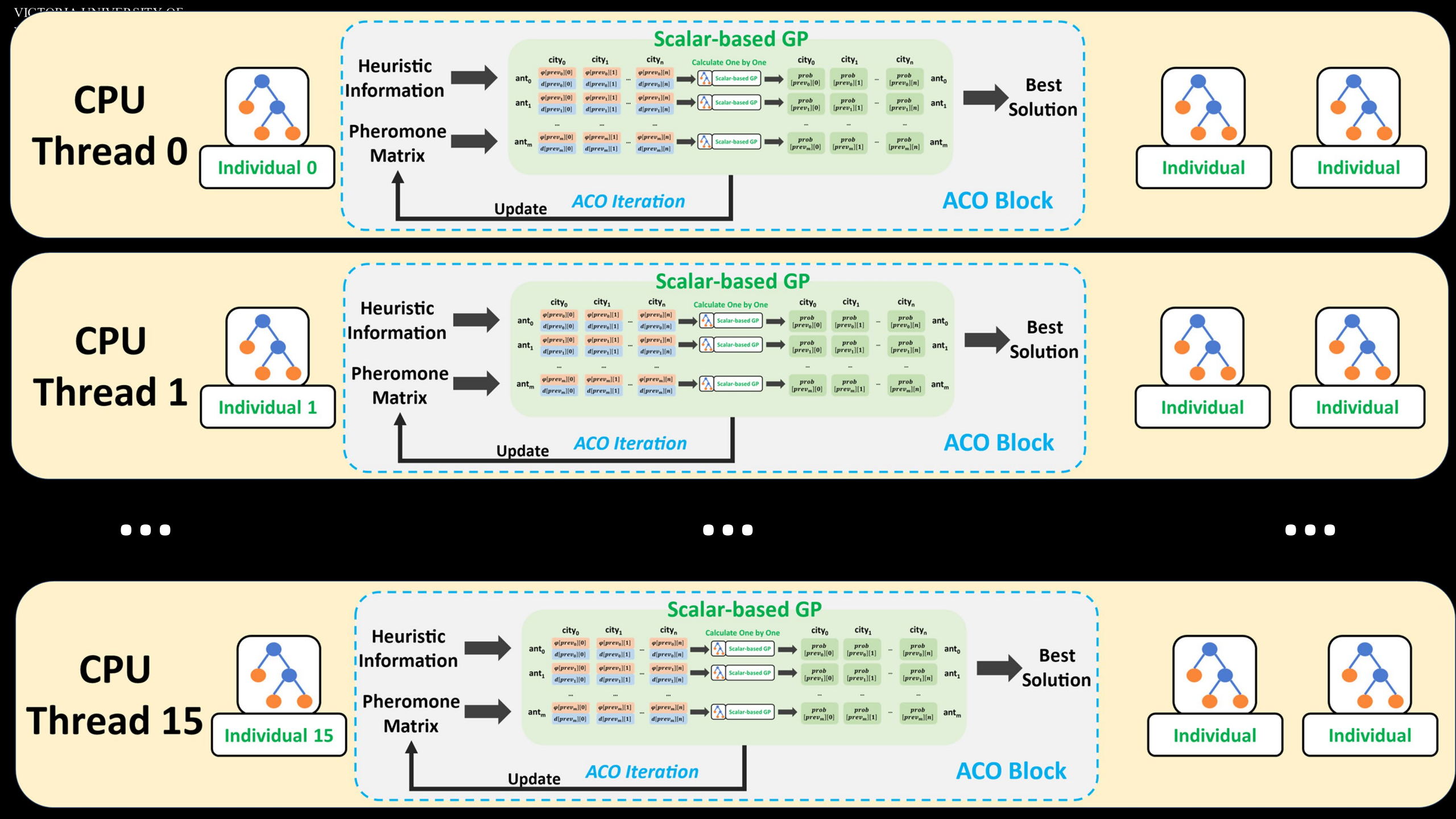
for *ant* **in** *all_ants*:
while *unvisited_cities* **is not empty:**
for *cities* **in** *unvisted_cities*:



VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

Parallel Simulation/Evaluation

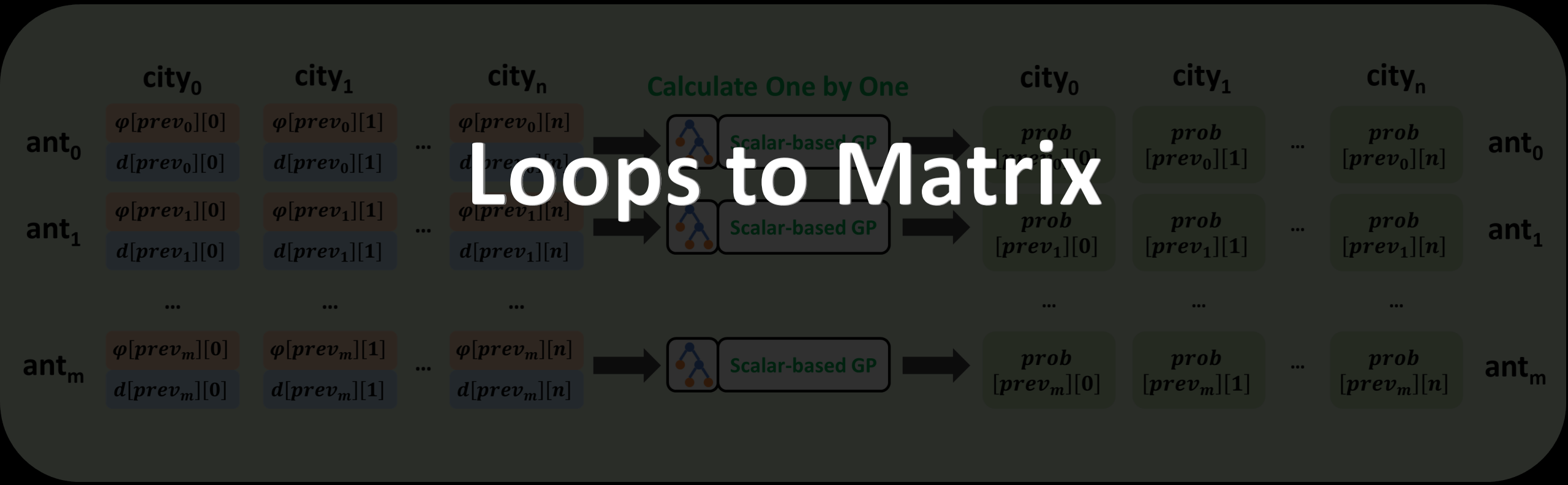


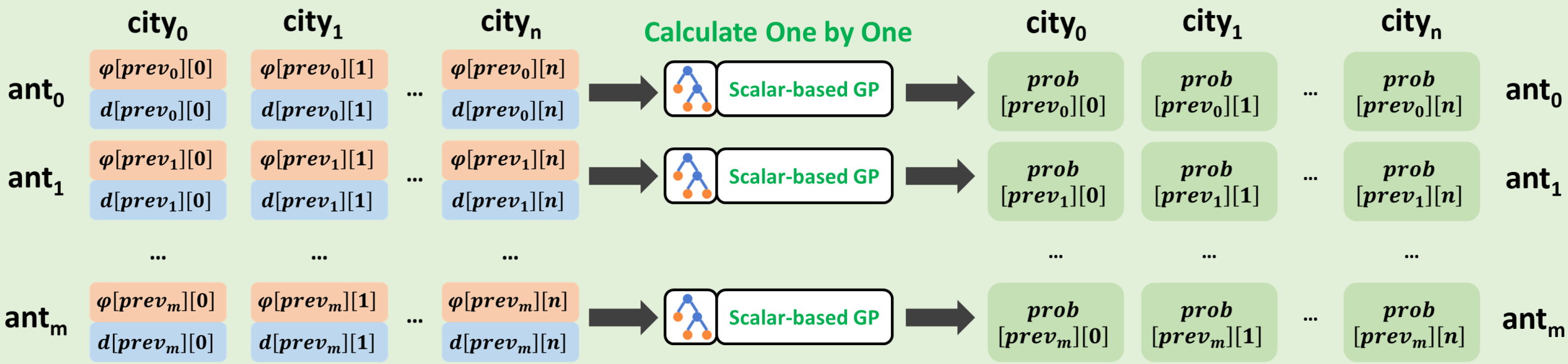

CPU Thread 0
Individual 0
Scalar-based GP
Heuristic Information
Pheromone Matrix
Calculate One by One
Scalar-based GP
Update
ACO Iteration
ACO Block
Best Solution
Individual
Individual
CPU Thread 1
Individual 1
Scalar-based GP
Heuristic Information
Pheromone Matrix
Calculate One by One
Scalar-based GP
Update
ACO Iteration
ACO Block
Best Solution
Individual
Individual
CPU Thread 15
Individual 15
Scalar-based GP
Heuristic Information
Pheromone Matrix
Calculate One by One
Scalar-based GP
Update
ACO Iteration
ACO Block
Best Solution
Individual
Individual

VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

```
for ant in all_ants:
    while unvisited_cities is not empty:
        for cities in unvisted_cities:
```

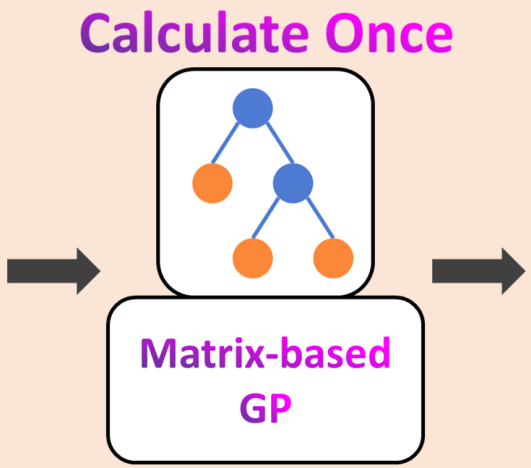
Loops to Matrix





$$\begin{bmatrix} \varphi[prev_0][0] & \cdots & \varphi[prev_0][n] \\ \vdots & \ddots & \vdots \\ \varphi[prev_m][0] & \cdots & \varphi[prev_0][n] \end{bmatrix}$$

$$\begin{bmatrix} d[prev_0][0] & \cdots & d[prev_0][n] \\ \vdots & \ddots & \vdots \\ d[prev_m][0] & \cdots & d[prev_0][n] \end{bmatrix}$$



$$\begin{bmatrix} prob[prev_0][0] & \cdots & prob[prev_0][n] \\ \vdots & \ddots & \vdots \\ prob[prev_m][0] & \cdots & prob[prev_0][n] \end{bmatrix}$$

```
for individual in population:
    #ACO simulation
    for ant in all_ants:
        while unvisited_cities is not empty:
            for cities in unvisted_cities:
                calculate_priority by individual;
                choose next_city;
            end for;
        end while;
    end for;
    offspring;
end for
```

VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

```
for individual in population:
    #ACO simulation
    for ant in all_ants:
        while unvisited_cities is not empty:
            for cities in unvisited_cities:
                calculate_priority by individual;
            choose next_city;
        end for;
    end while;
end for;
offspring;
end for
```

How to deal with *unvisited_cities* ?

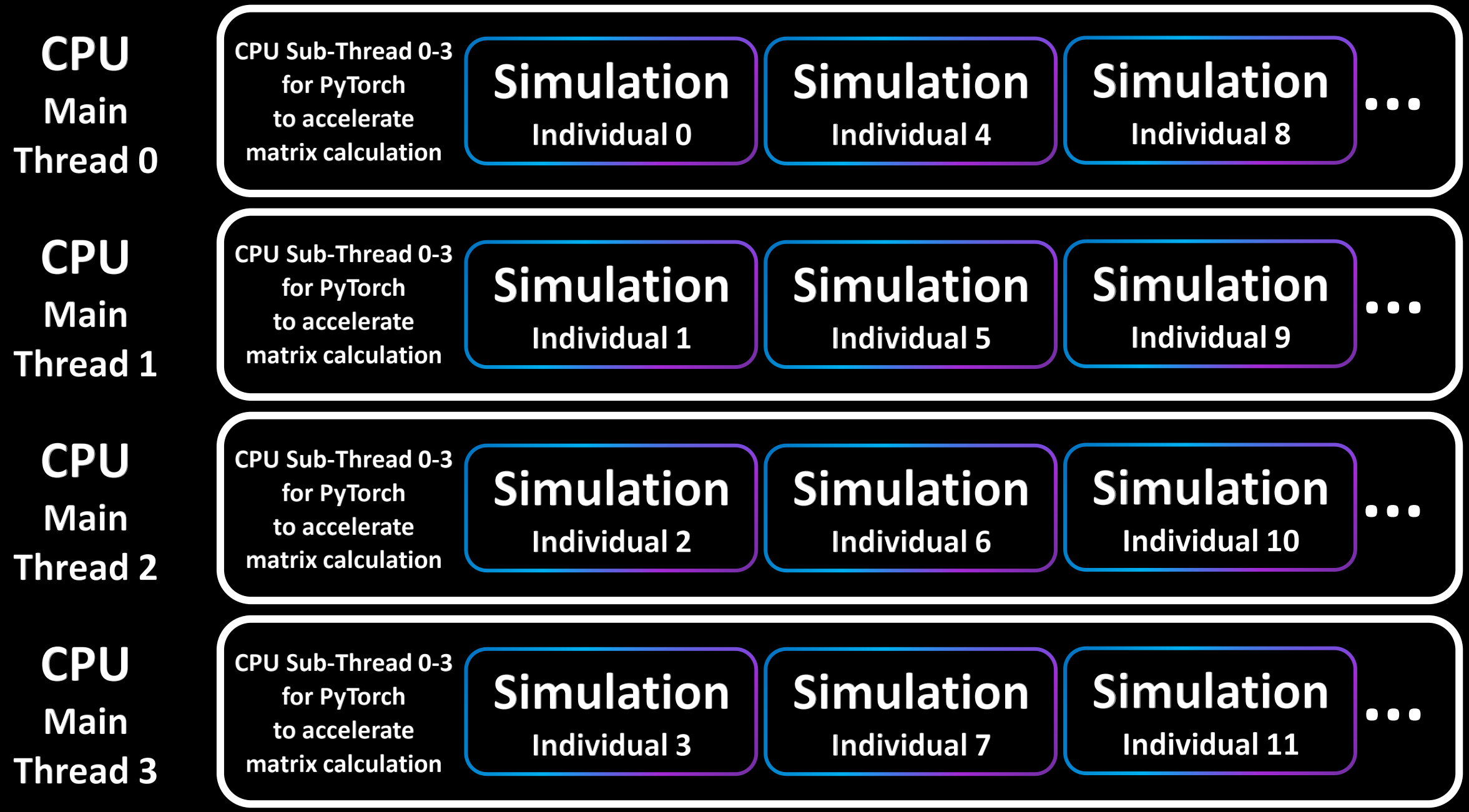
VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

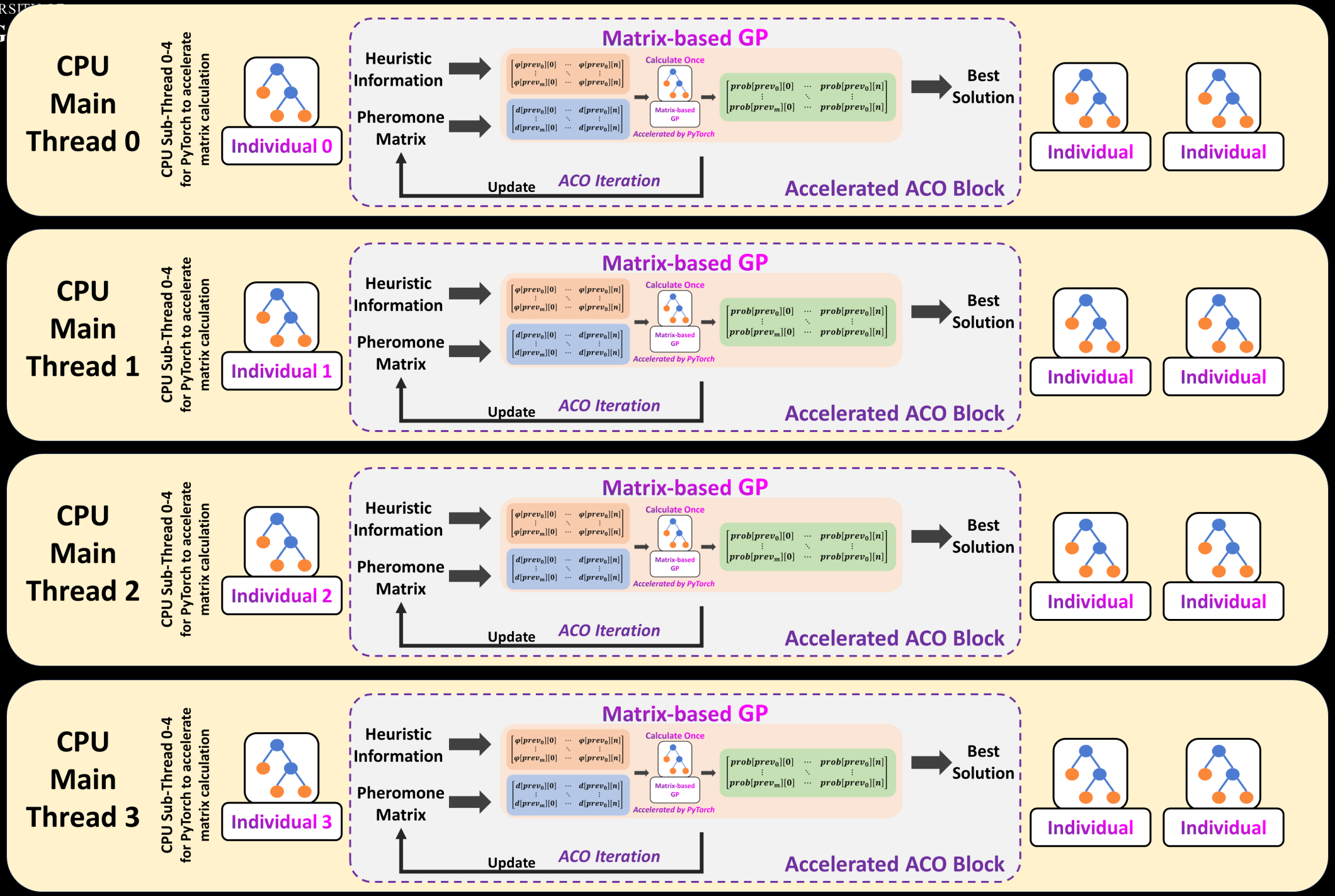
```
for individual in population:
    #ACO simulation
    for ant in all_ants:
        while unvisited_cities is not empty:
            for cities in unvisted_cities:
                calculate_priority by individual;
                choose next_city;
            end for;
        end while;
    end for;
    offspring;
end for
```

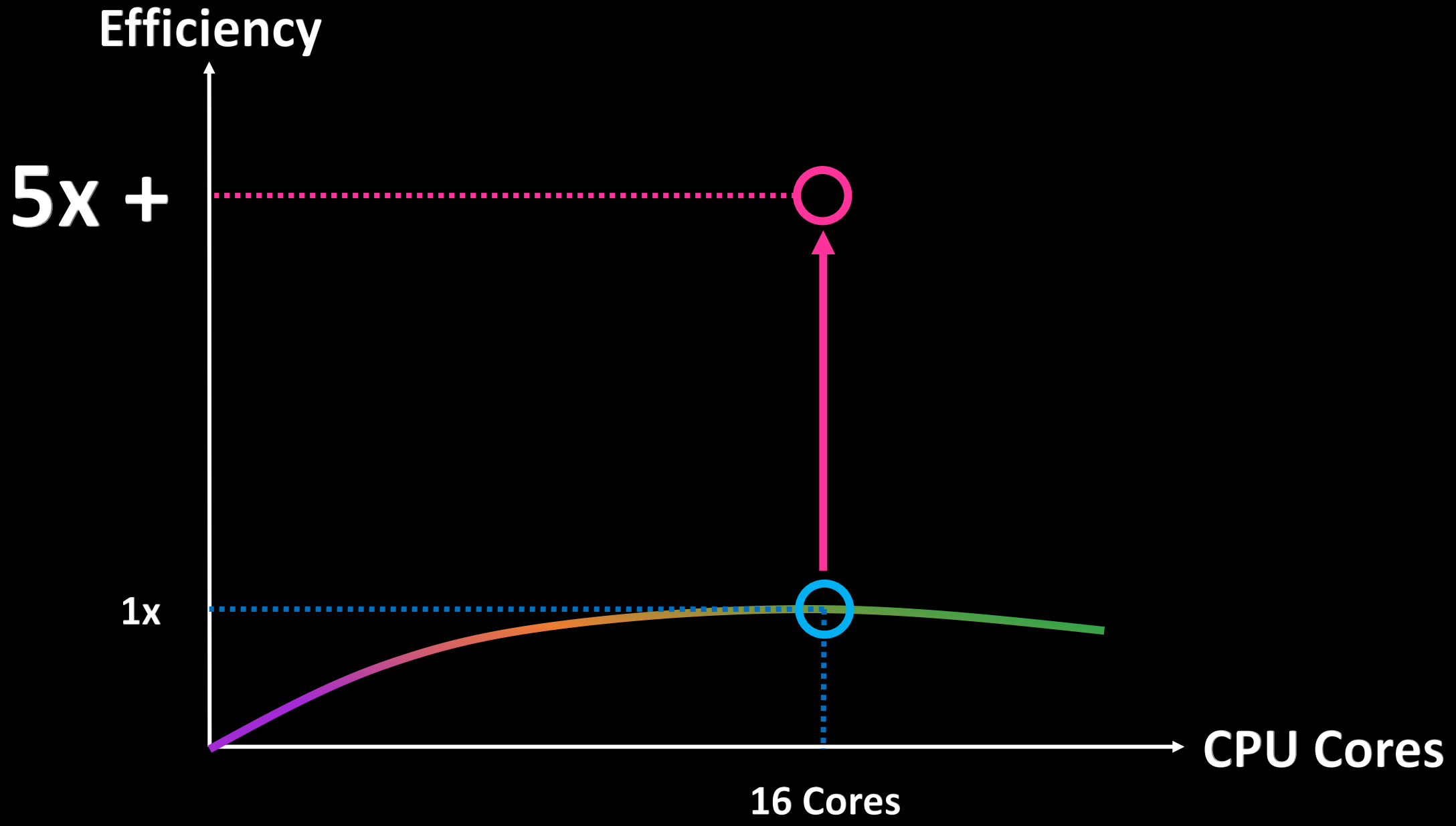
Matrix A × Matrix B × Mask

VICTORIA UNIVERSITY OF WELLINGTON
TE HERENGA WAKA

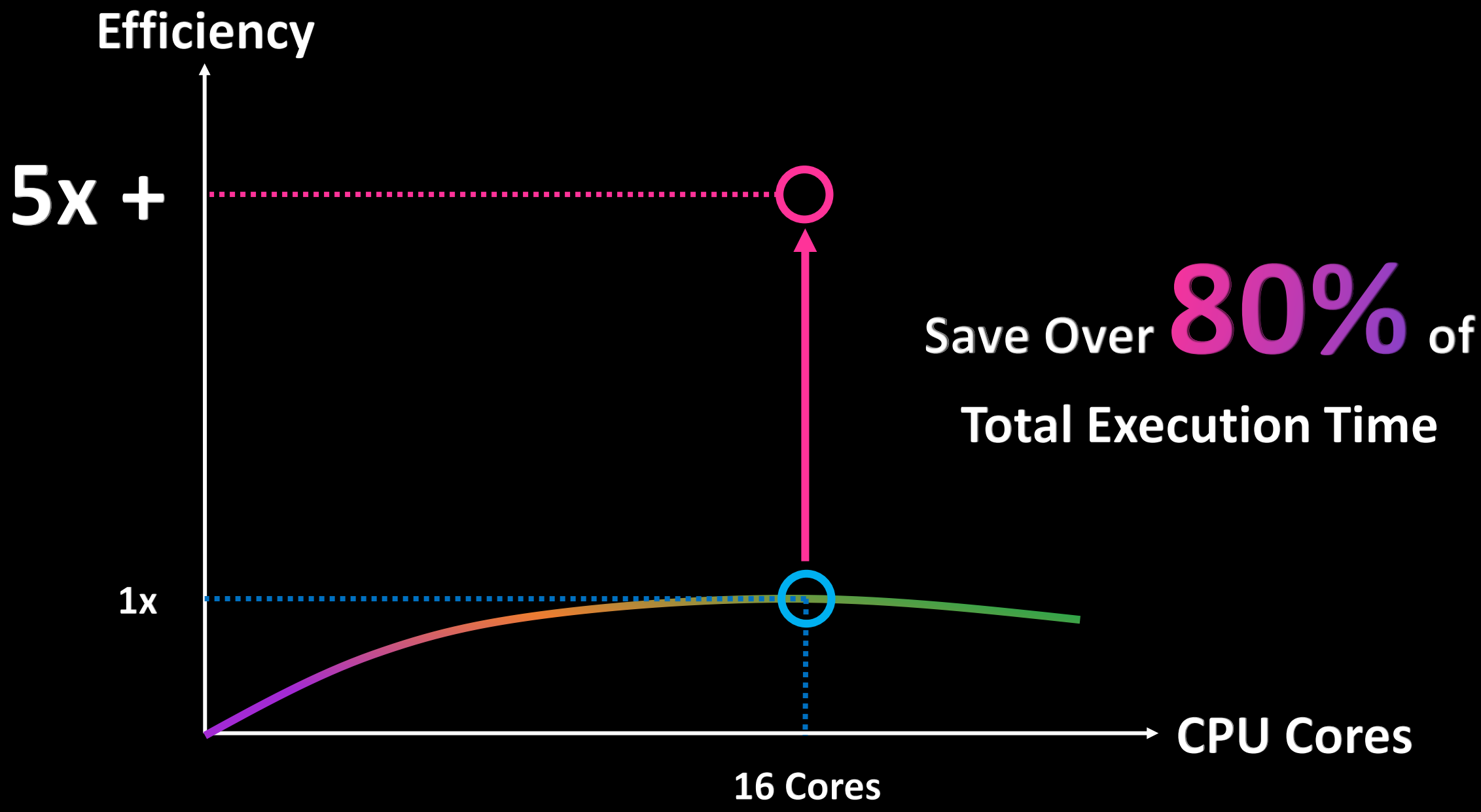
Parallel inside Simulation/Evaluation






Efficiency
5x +
1x
CPU Cores
16 Cores

VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA



Save Over **80%** of

Total Execution Time

Interesting Blogs and GitHub Projects

How to accelerate matrix calculation

· https://github.com/flame/how-to-optimize-gemm/wiki
· http://technicaldiscovery.blogspot.com/2011/06/speeding-up-python-numpy-cython-and.html
· https://scipy.github.io/old-wiki/pages/PerformancePython.html#An_implementation_in_C.2B-.2B-

Interesting Blogs and GitHub Projects

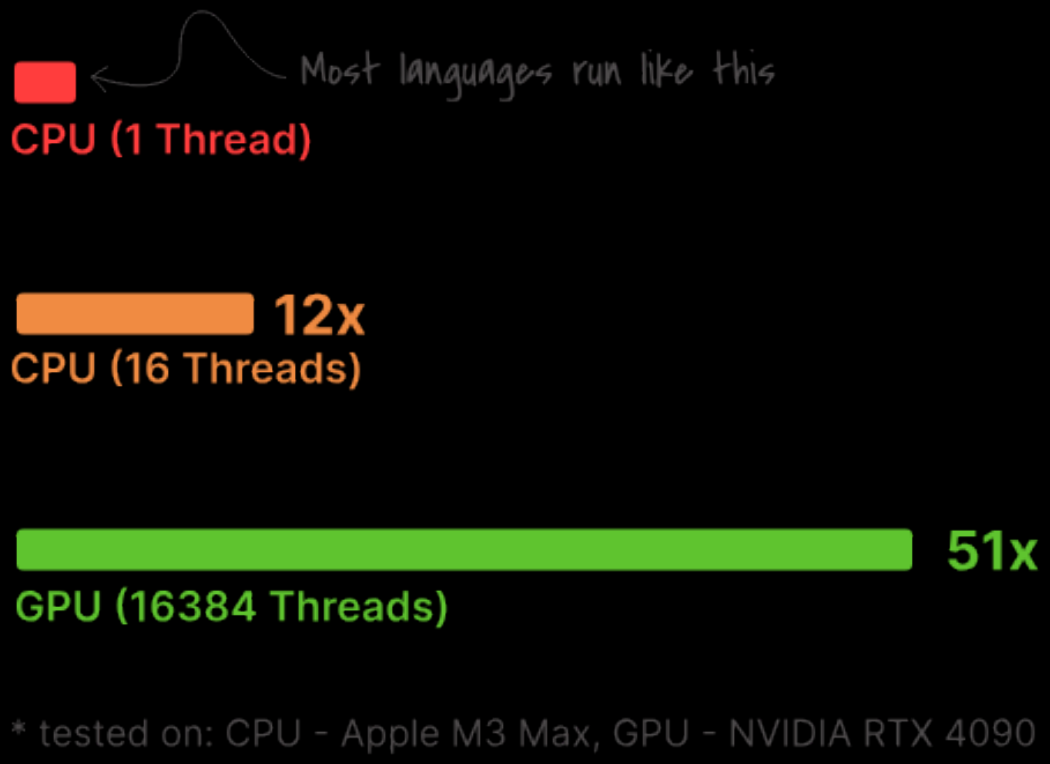
Implementing PyTorch Operators with CUDA

· https://zhuanlan.zhihu.com/p/595851188?utm_campaign=shareopn&utm_content=group3_article&utm_medium=social&utm_psn=1779857889484021760&utm_source=wechat_session
· https://github.com/YuxueYang1204/CudaDemo/tree/main

Interesting Blogs and GitHub Projects

Bend : A PARALLEL LANGUAGE on CUDA, feels just like Python

https://higherorderco.com/



VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

Any Questions?

Thank you for your listening!

Bocheng Lin 2024.6.5