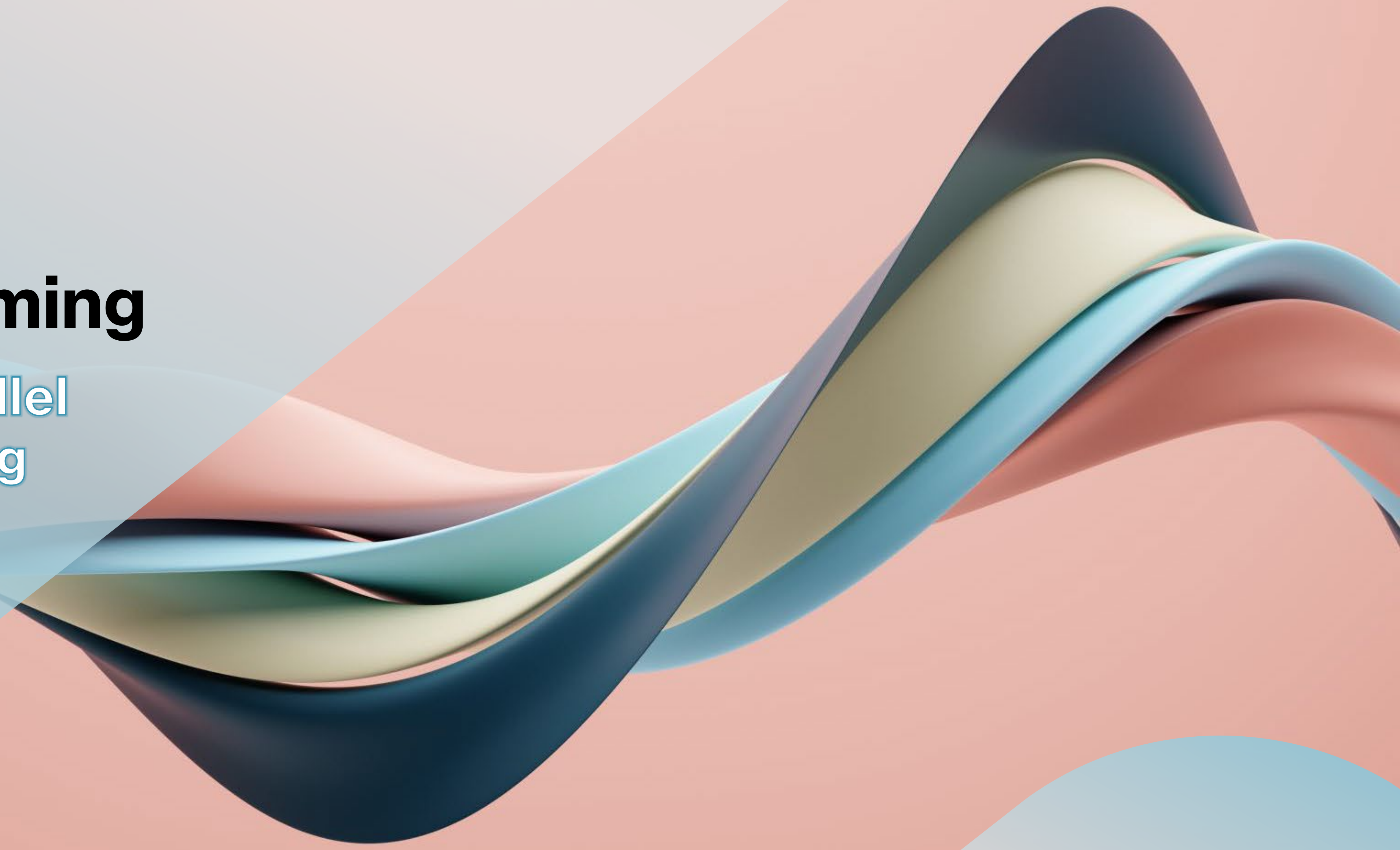
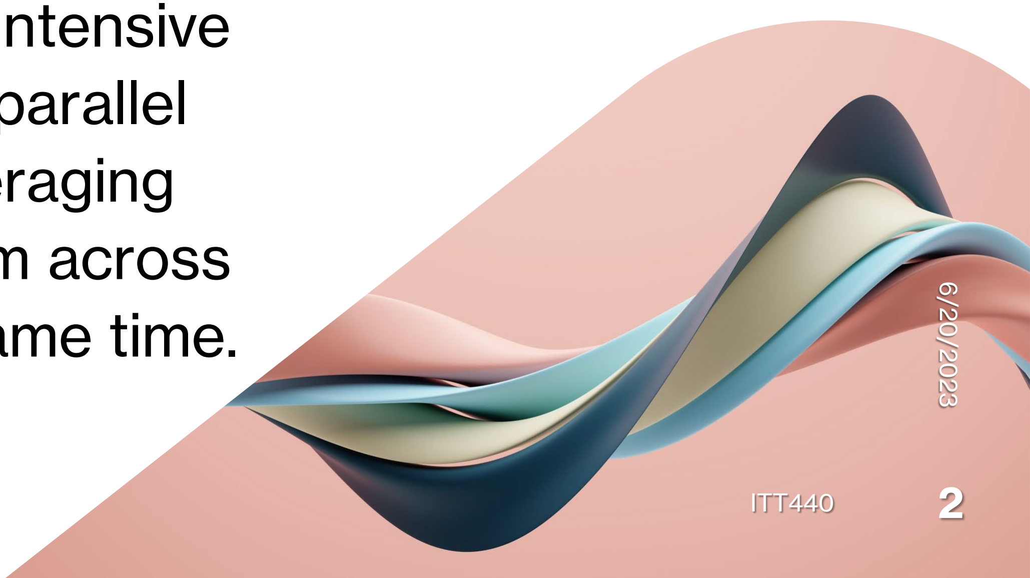
ITT440 Network Programming

Python Parallel Programming



Basic Knowledge

- Complex problems require complex solutions.
- Instead of waiting hours for a program to finish running, why not utilize parallel programming?
- Parallel programming helps developers break down the tasks that a program must complete into smaller segments of work that can be done in parallel.
- While parallel programming can be a more time intensive effort up front for developers to create efficient parallel algorithms and code, it overall saves time by leveraging parallel processing power by running the program across multiple compute nodes and CPU cores at the same time.



Example

```
import multiprocessing
import time
def square(x):
        return x * x

if __name__ == '__main__':
        pool = multiprocessing.Pool()
        result_async = [pool.apply_async(square, args = (i, )) for i in range(10)]
        results = [r.get() for r in result_async]
        print("Output: {}".format(results))
```

Exercise

Write a Parallel Programming in Python to calculate cube number from 1 until 1 000 000

References

- https://totalview.io/blog/what-is-parallel-programming
- https://www.geeksforgeeks.org/parallel-processing-in-python/