

Concurrency Theory - Fall 2020

(Course code: DM861)

Fabrizio Montesi



DEPARTMENT OF MATHEMATICS
AND COMPUTER SCIENCE



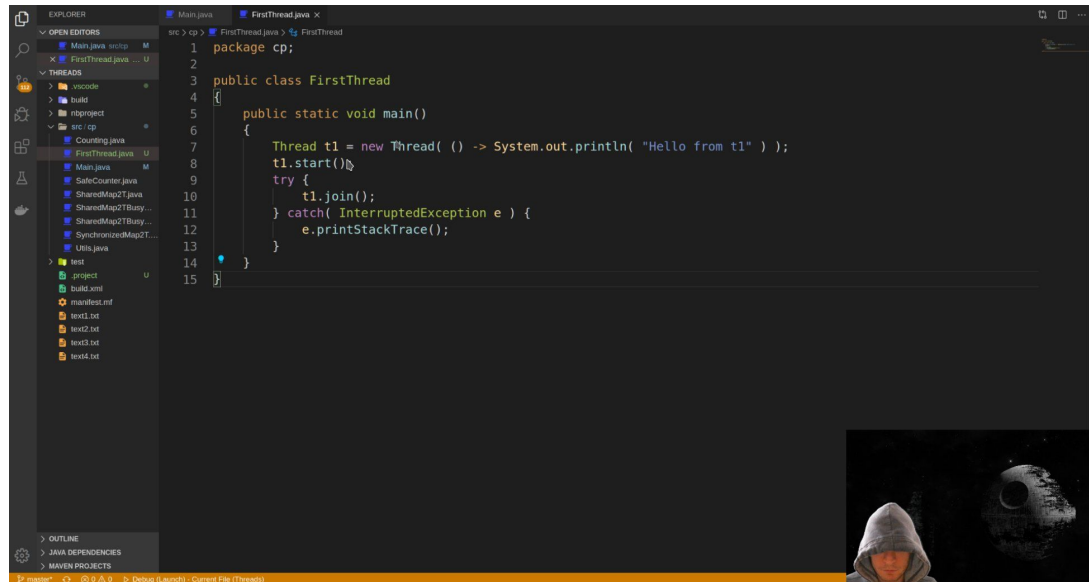
fabriziomontesi.com



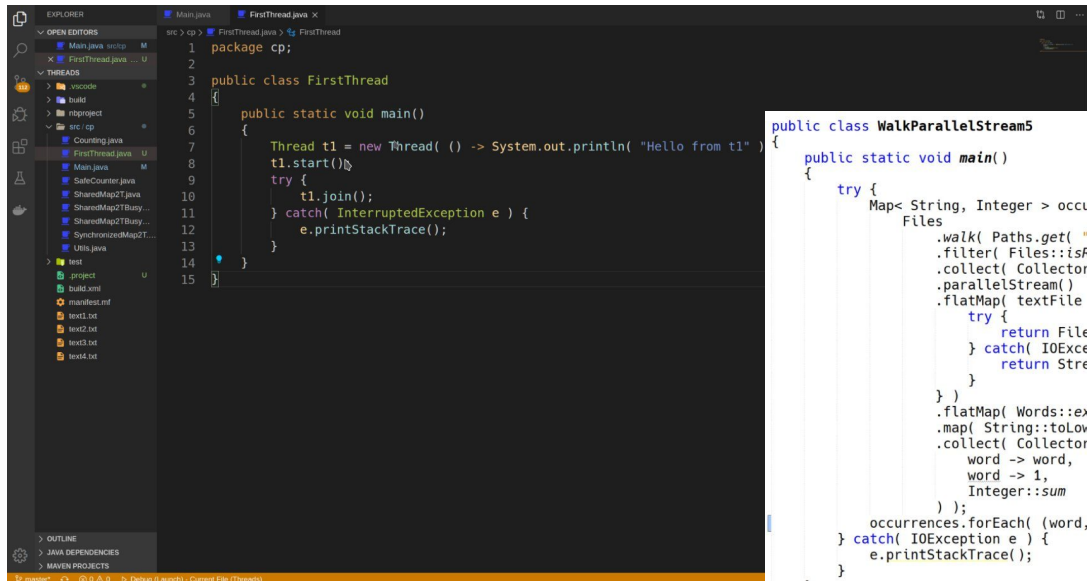
twitter.com/famontesi

You might know me from
DM563: Concurrent Programming

You might know me from DM563: Concurrent Programming



You might know me from DM563: Concurrent Programming



```
package cp;

public class FirstThread
{
    public static void main()
    {
        Thread t1 = new Thread( () -> System.out.println( "Hello from t1" ) );
        t1.start();
        try {
            t1.join();
        } catch( InterruptedException e ) {
            e.printStackTrace();
        }
    }
}
```

```
public class WalkParallelStream5
{
    public static void main()
    {
        try {
            Map< String, Integer > occurrences =
                Files
                    .walk( Paths.get( "data" ) )
                    .filter( Files::isRegularFile )
                    .collect( Collectors.toList() )
                    .parallelStream()
                    .flatMap( textFile -> {
                        try {
                            return Files.lines( textFile );
                        } catch( IOException e ) {
                            return Stream.empty();
                        }
                    } )
                    .flatMap( Words::extractWords )
                    .map( String::toLowerCase )
                    .collect( Collectors.toMap(
                        word -> word,
                        word -> 1,
                        Integer::sum
                    ) );
            occurrences.forEach( (word, n) -> System.out.println( word + ": " + n ) );
        } catch( IOException e ) {
            e.printStackTrace();
        }
    }
}
```



Who is this for?

- Computer Scientists
- Applied mathematicians

What are we going to study?

What are we going to study?

Concurrency, of course!

What are we going to study?

Concurrency, of course!

- Systems where multiple tasks run at the same time.

What are we going to study?

Concurrency, of course!

- Systems where multiple tasks run at the same time.
- Systems with multiple participants.

Why?

Why?

- Concurrency is everywhere.

Why?

- Concurrency is everywhere.
- The future is more and more concurrent.

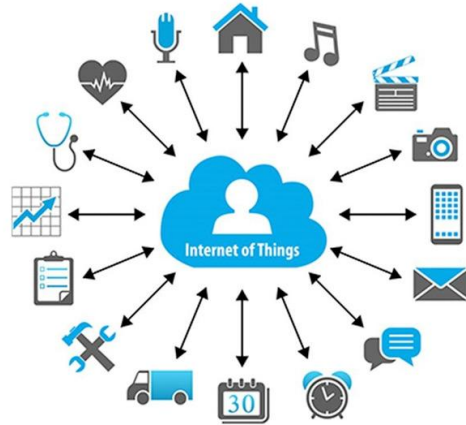
Why?

- Concurrency is everywhere.
- The future is more and more concurrent.
- New cool tools become obsolete very quickly.

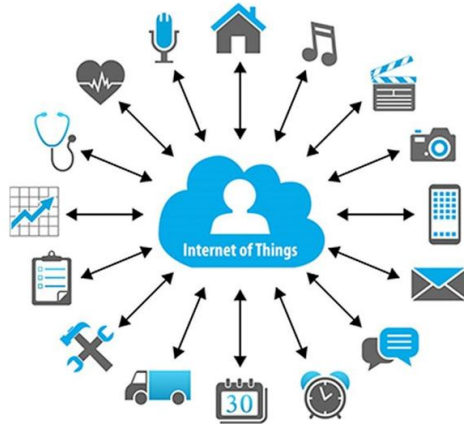
Why?

- Concurrency is everywhere.
- The future is more and more concurrent.
- New cool tools become obsolete very quickly.
- Need to understand reusable principles.

Computer networks



Computer networks



NEM ID ? | X

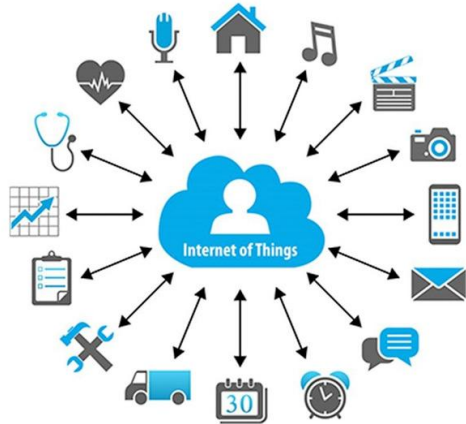
NemID

User ID ☐ Remember me

Password [Forgot password?](#)

Accept

Computer networks



NEM ID ? | X

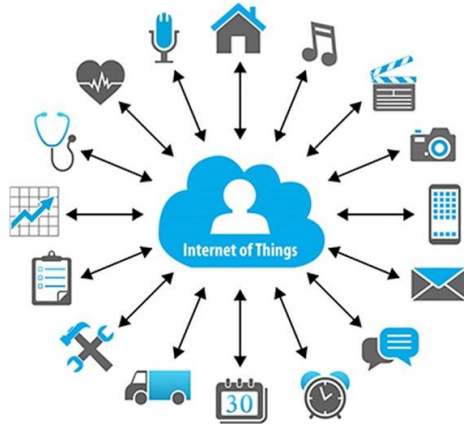
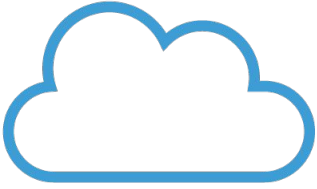
NemID

User ID ☐ Remember me

Password [Forgot password?](#)

Accept

Computer networks



eScience



NEM ID

?

×

NemID

User ID

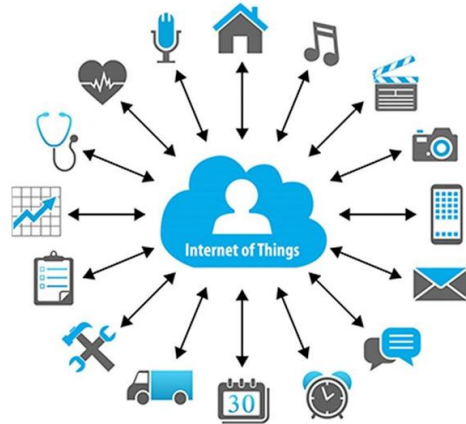
☐ Remember me

Password

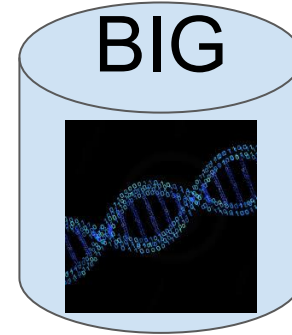
[Forgot password?](#)

Accept

Computer networks



eScience



NEM ID ? | X

NemID

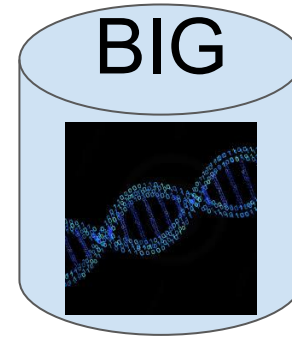
User ID ☐ Remember me

Password [Forgot password?](#)

Computer networks



eScience



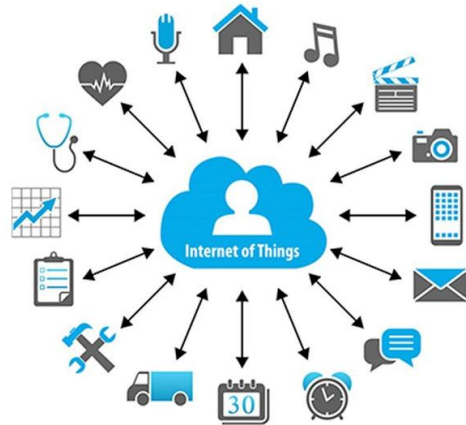
NEM ID ? | X

NemID

User ID ☐ Remember me

Password [Forgot password?](#)

Accept



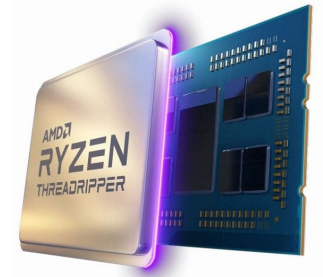
Multi-core processors



2 or 4 cores



8 cores

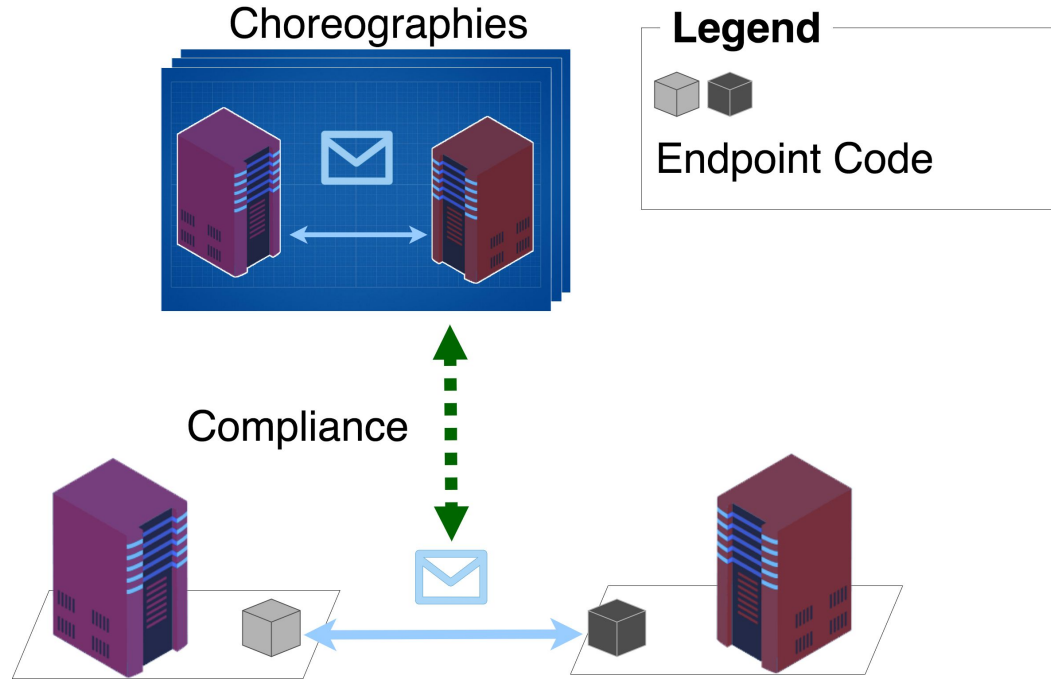


64 cores

What is this relevant for?

- Lots of settings, including:
 - software development;
 - cloud computing, edge computing, Internet of Things, microservices;
 - design of complex architectures (HW and SW);
 - models for bioinformatics/cheminformatics.

Choreography compliance



(Credits: Giallorenzo et al., 2020)

What's the course like?

- Frontal lectures with blackboard and discussion on definitions, proofs, etc.
- Exercises in groups or alone.
Teacher gives hints and assistance.

Exam and assessment

- Obligatory assignment during the course: solutions to 6 exercises from the lecture notes (easy if you follow exercise classes).
- Exam: written exam, pass/fail.

How do the content and the exam look like?

- Please visit the page from last year's course:

<https://www.fabriziomontesi.com/teaching/ct-2019/index.html>

- You will find:
 - all lecture notes (material 2);
 - an exam example (material 6).

Looking forward to meeting you!