

# Module 06:

## "Data Parallelism in TPL"



**TEKNOLOGISK**  
**INSTITUT**

# Agenda

- ▶ Introducing the Task Parallel Library
- ▶ The Parallel Class
- ▶ Parallel LINQ

# Task Parallel Library

- ▶ Task Parallel Library (TPL)
  - Was introduced in .NET 4.0
  - Enhanced in .NET 4.5
    - Special keywords are included in C# 5.0, C# 8.0
- ▶ Features
  - Task Parallelism
  - **Data Parallelism**
  - **Parallel LINQ**
  - Thread-safe collections
- ▶ Emerging trends leverage parallelism! Also .NET!

# Agenda

- ▶ Introducing the Task Parallel Library
- ▶ **The Parallel Class**
- ▶ Parallel LINQ

# The Parallel Class

- ▶ The **Parallel** class leverages data parallelism
- ▶ Provides a high-level API to data parallelism
  - Essentially abstracting thread pool coordination
  - A perfect fit for similar computations where data varies
- ▶ Note: No direct C# language support!

# The Parallel Class

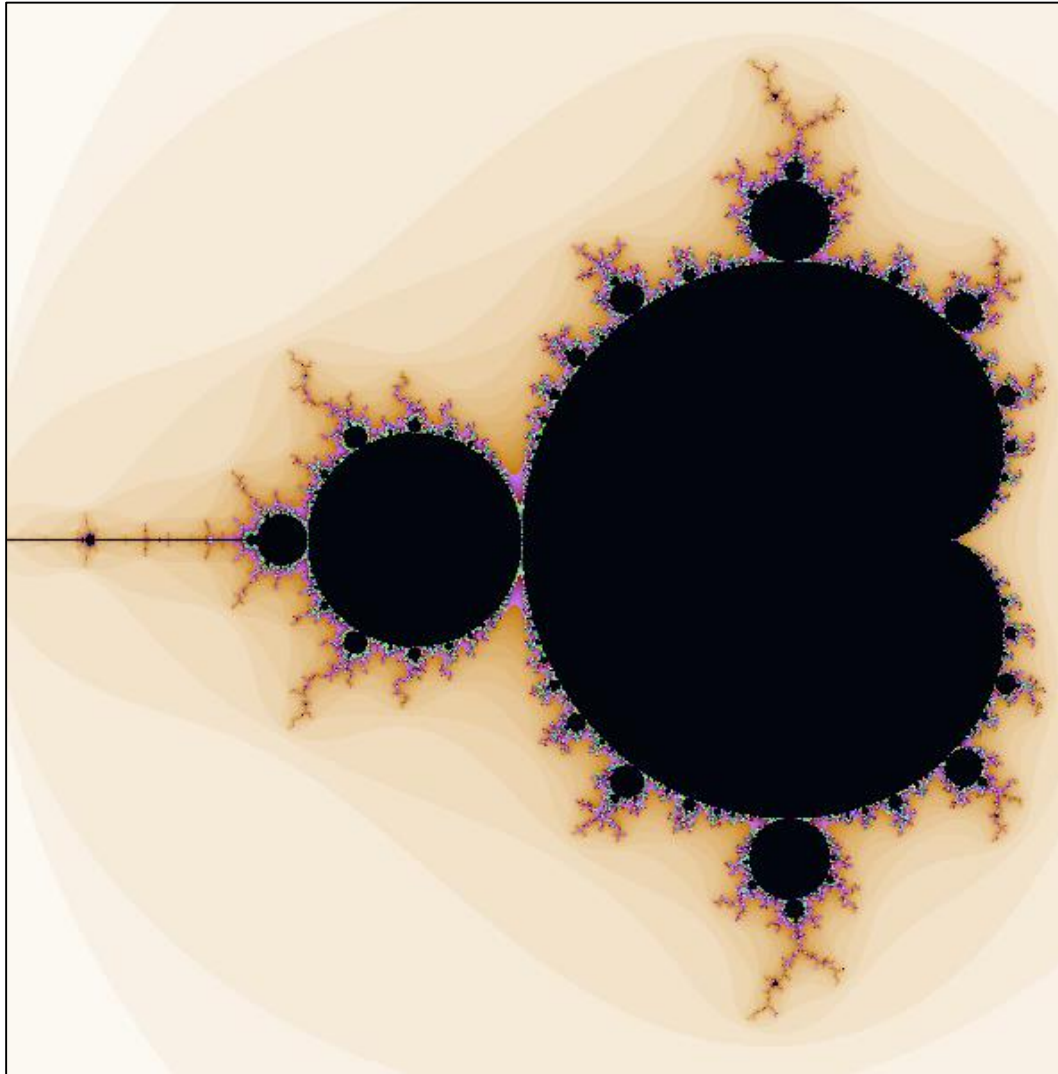
## ▶ Parallel.

- **Invoke()** invokes actions in parallel
- **For()** is a parallel **for**-loop
- **ForEach()** is a parallel **foreach**-loop

```
Parallel.For( 0, 1000, i =>  
    Console.WriteLine( $"Executing number {i,4}..." )  
);
```

- ▶ Developer's responsibility that iterations are in fact independent

# Example: The Mandelbrot Set



# Additional Parallel Options

- ▶ Options and refinements are provided through various overloads
  - The **ParallelLoopState** and **ParallelLoopResult** classes

```
ParallelLoopResult result = Parallel.For( 0, 1000, ( i, state ) =>
{
    if( i == 87 )
    {
        state.Break();
    }
    ...
}
```

- The **ParallelOptions** class
  - **MaxDegreeOfParallelism**
  - **CancellationToken**



# Agenda

- ▶ Introducing the Task Parallel Library
- ▶ The Parallel Class
- ▶ **Parallel LINQ**

# Introducing Parallel LINQ

- ▶ Parallel LINQ
  - Implements the well-know API of LINQ
  - Use thread pool to (potentially) evaluate queries in parallel
  
- ▶ (Almost) all operations have parallel implementations
  - Filtering
  - Sorting
  - Grouping
  - Aggregation (\*)
  - ...

# Using Parallel LINQ

- ▶ Parallel LINQ = Parallel LINQ
  - **ParallelEnumerable** class is defined in **System.Linq** namespace
- ▶ **ParallelEnumerable**
  - **AsParallel<T>()**
  - **AsSequential<T>()**
  - **WithCancellation<T>()**
  - **WithDegreeOfParallelism<T>()**

```
var even = from i in numbers.AsParallel()  
           .WithCancellation( token )  
           .WithDegreeOfParallelism( 2 )  
           where i % 2 == 0  
           select i;
```

# Best Practices for Using PLINQ

## ▶ Do...

- Use for low-hanging, inherently parallelizable CPU-bound computations
- Remember that UI updates from Parallel need dispatching

## ▶ Don't...

- Automatically always use PLINQ (due to potential overhead)
- Use for out-of-process or IO-bound tasks
  - SQL access
  - Entity Framework

# Summary

- ▶ Introducing the Task Parallel Library
- ▶ The Parallel Class
- ▶ Parallel LINQ



WINCUBATE

Jesper Gulmann Henriksen

PhD, MCT, MCSD, MCPD

Phone : +45 22 12 36 31

Email : [jgh@wincubate.net](mailto:jgh@wincubate.net)

WWW : <http://www.wincubate.net>

Ringgårdsvej 4A

8270 Højbjerg

Denmark