Parallel programming in the .NET Framework

Tang Gabriel
1593362

Khau Stéphane
1593361

Zheng Arnaud
1593365

Charotte Steeve
1593367

Barroqueiro Thomas
1593363

Hello!

We are group 4

We are here to make a presentation about
Parallel programming in the .NET Framework

Summary

Background

Terminologies

Functionalities

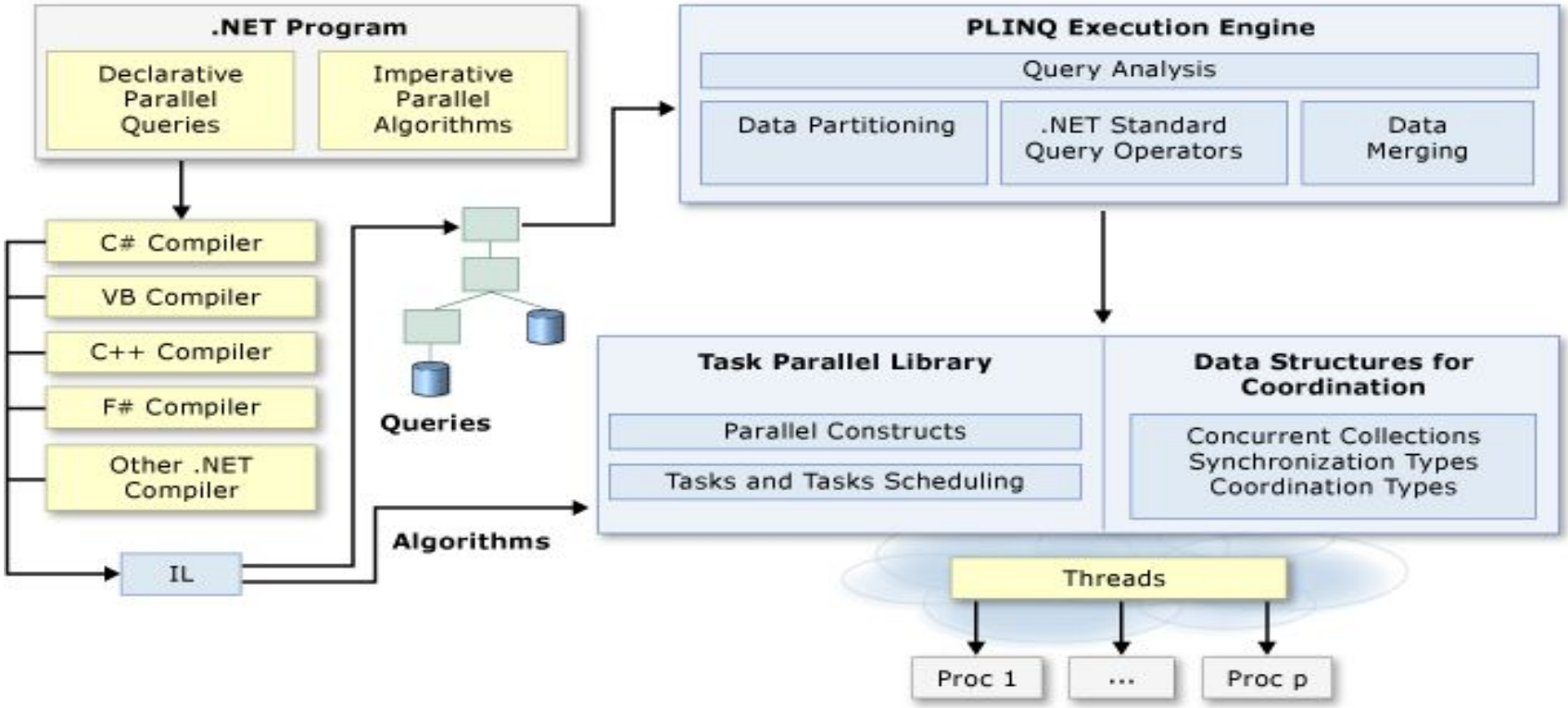
Usage

Background

Functionalities in few words

Many personal computers and workstations have two or four cores (that is, CPUs) that enable multiple threads to be executed simultaneously

Schema


.NET Program
Declarative Parallel Queries
Imperative Parallel Algorithms
C# Compiler
VB Compiler
C++ Compiler
F# Compiler
Other .NET Compiler
IL
Queries
Algorithms
PLINQ Execution Engine
Query Analysis
Data Partitioning
.NET Standard Query Operators
Data Merging
Task Parallel Library
Parallel Constructs
Tasks and Tasks Scheduling
Data Structures for Coordination
Concurrent Collections
Synchronization Types
Coordination Types
Threads
Proc 1
...
Proc p

Terminologies

Terminologies

- Multithreading
  - Each thread defines a unique flow of control
  - Provides better user experience by optimizing calculations
- Multiprocessing
  - Each driver of a module runs its own child process
  - Separates each module from the others
- Multitasking
  - Having several applications running and working at the same time

Functionalities

Functionalities

- Parallel Programing
  - Multi-core machine power usage
  - It's about how to partition a single piece of work into multiple concurrent units.

Parallel Programing

- It’s composed by
  - The Task Parallel Library (TPL)
    - Task Class
    - Parallel Class
      - For
      - Foreach
      - Invoke
  - Parallel LINQ (PLINQ)
    - Built on top of the TPL and exposes the familiar LINQ as Parallel extensions

4

Usage

Usage

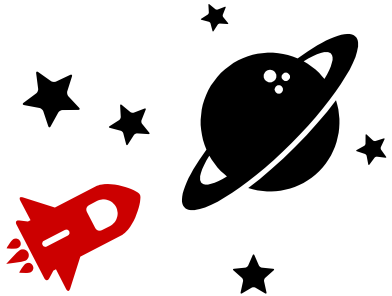
- Task Parallel Library (TPL)
- Task Schedulers and Synchronization Context
- Task Cancellation

Usage

```
== Classic ==
root 1 : 49999995000000 - 594 milliseconds
root 2 : 21081849486,44 - 1169 milliseconds
root 3 : 1615825909,52 - 1791 milliseconds
root 4 : 449873031,71 - 2372 milliseconds
root 5 : 209323856,38 - 2988 milliseconds
root 6 : 125811358,09 - 3601 milliseconds
root 7 : 87499994,61 - 4217 milliseconds
root 8 : 66657258,9 - 4792 milliseconds
root 9 : 53953579,12 - 5401 milliseconds
root 10 : 45562472,86 - 6009 milliseconds
root 11 : 39680309,15 - 6624 milliseconds
root 12 : 35364799,34 - 7240 milliseconds
root 13 : 32083137 - 7846 milliseconds
root 14 : 29514589,47 - 8456 milliseconds
root 15 : 27456040,89 - 9069 milliseconds
root 16 : 25773359,45 - 9648 milliseconds
root 17 : 24374801,7 - 10261 milliseconds
root 18 : 23195714,87 - 10870 milliseconds
root 19 : 22189352,33 - 11473 milliseconds
=== Total time 11474 milliseconds ===
```

```
== Parallel ==
root 9 : 53953579,12 - 858 milliseconds
root 3 : 1615825909,52 - 860 milliseconds
root 17 : 24374801,7 - 862 milliseconds
root 11 : 39680309,15 - 864 milliseconds
root 7 : 87499994,61 - 1038 milliseconds
root 15 : 27456040,89 - 1091 milliseconds
root 13 : 32083137 - 1094 milliseconds
root 5 : 209323856,38 - 1120 milliseconds
root 1 : 49999995000000 - 1216 milliseconds
root 4 : 449873031,71 - 1692 milliseconds
root 10 : 45562472,86 - 1716 milliseconds
root 18 : 23195714,87 - 1720 milliseconds
root 12 : 35364799,34 - 1724 milliseconds
root 8 : 66657258,9 - 2006 milliseconds
root 16 : 25773359,45 - 2044 milliseconds
root 14 : 29514589,47 - 2060 milliseconds
root 6 : 125811358,09 - 2104 milliseconds
root 2 : 21081849486,44 - 2117 milliseconds
root 19 : 22189352,33 - 2374 milliseconds
=== Total time 2375 milliseconds ===
```

```
== Synchronization ==
root 20 : 21081849486,44 - 830 milliseconds
root 20 : 449873031,71 - 836 milliseconds
root 20 : 49999995000000 - 847 milliseconds
root 20 : 125811358,09 - 859 milliseconds
root 20 : 209323856,38 - 861 milliseconds
root 20 : 87499994,61 - 863 milliseconds
root 20 : 66657258,9 - 895 milliseconds
root 20 : 1615825909,52 - 924 milliseconds
root 20 : 45562472,86 - 1700 milliseconds
root 20 : 29514589,47 - 1721 milliseconds
root 20 : 53953579,12 - 1728 milliseconds
root 20 : 39680309,15 - 1740 milliseconds
root 20 : 32083137 - 1741 milliseconds
root 20 : 25773359,45 - 1771 milliseconds
root 20 : 27456040,89 - 1787 milliseconds
root 20 : 35364799,34 - 1803 milliseconds
root 20 : 24374801,7 - 2416 milliseconds
root 20 : 22189352,33 - 2437 milliseconds
root 20 : 23195714,87 - 2445 milliseconds
=== Total time 2446 milliseconds ===
```



Demo Time

Thanks!