Chapter 22. Parallel Programming

In this chapter, we cover the multithreading APIs and constructs aimed at leveraging multicore processors:

- Parallel LINQ or *PLINQ*
- The `Parallel` class
- The *task parallelism* constructs
- The *concurrent collections*

These constructs are collectively known (loosely) as Parallel Framework (PFX). The `Parallel` class together with the task parallelism constructs is called the *Task Parallel Library* (TPL).

You'll need to be comfortable with the fundamentals in Chapter 14 before reading this chapter—particularly locking, thread safety, and the `Task` class.

> **NOTE**
>
> .NET offers a number of additional specialized APIs to help with parallel and asynchronous programming:
>
> - `System.Threading.Channels.Channel` is a high-performance asynchronous producer/consumer queue, introduced in .NET Core 3.
> - *Microsoft Dataflow* (in the `System.Threading.Tasks.Dataflow` namespace) is a sophisticated API for creating networks of buffered *blocks* that execute actions or data transformations in parallel, with a semblance to actor/agent programming.
> - *Reactive Extensions* implements LINQ over `IObservable` (an alternative abstraction to `IAsyncEnumerable`) and excels at combining asynchronous streams. Reactive extensions ships in the *System.Reactive* NuGet package.