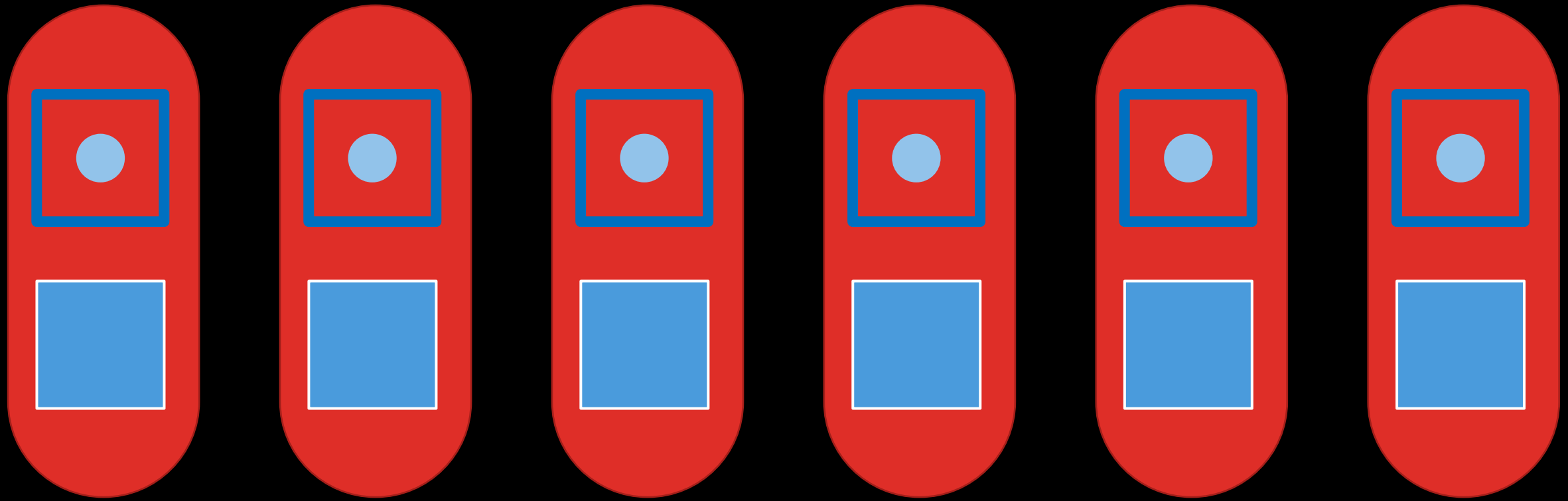
Better Parallel Code with C# Channels

Jeremy Clark
www.jeremybytes.com
@jeremybytes

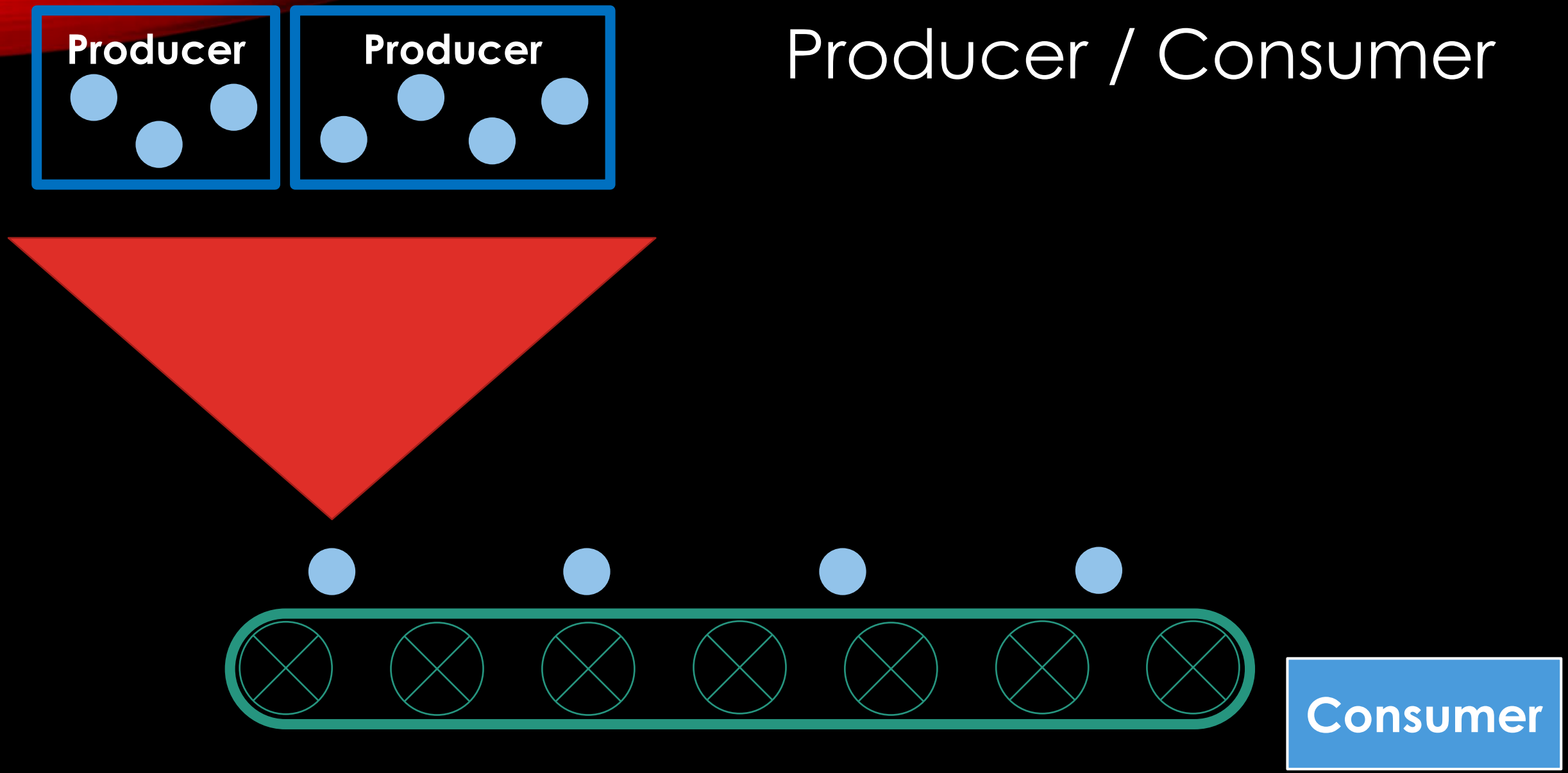
Channel<T>

- Rolled in to .NET Core 3.0
  Prior to that, there was a separate NuGet package
- Communicate between async methods/functions
- Similar to a concurrent queue, but with reader/writer optimizations and signaling built-in
- Not for communicating between distributed applications

Get Data / Use Data



Producer / Consumer



Producers and consumers can communicate asynchronously through a channel.

Major Operations

- Channel<T>
  - Create a channel
  - Write to a channel
  - Read from a channel
  - Mark the channel “complete”

Creating a Channel

- CreateBounded<T>
  - Creates a channel of a specific size
  - If the channel is full, writers are blocked until space is available

```
var channel = Channel.CreateBounded<Person>(10);
```

Channel Reader / Writer

Reader property

- ChannelReader<T>
  - ReadAllAsync()

Writer property

- ChannelWriter<T>
  - WriteAsync()
  - Complete()

Writing to a Channel

- writer.WriteAsync()
  - Writes an item to the channel

```
await writer.WriteAsync(item);
```

Marking a Channel "Complete"

- writer.Complete()
  - Indicates that no further items will be written
  - Writing to a "complete" channel throws an exception
  - Reading from a "complete" channel will continue normally until the channel is empty

Reading from a Channel

- reader.ReadAllAsync()
  - Returns an IAsyncEnumerable<T>

```
await foreach (var item in reader.ReadAllAsync())
{
    // use item here
}
```

  - If the channel is empty, the loop will pause until an item is available.
  - If the channel is "complete", the loop will exit.

Other Stuff

- ChannelReader<T>
  - WaitToReadAsync()
  - ReadAsync()
  - TryRead()
- ChannelWriter<T>
  - WaitToWriteAsync()
  - TryWrite()
  - TryComplete()

Other Stuff

- Channel.CreateUnbounded<T>

- ChannelCreationOptions
  - SingleReader
  - SingleWriter
  - AllowSynchronousContinuations
    These allow for compiler and runtime optimizations

Comparing Loops

	Await	Task	Channel	ForEachAsync
Runs in Parallel	No	**Yes**	**Yes**	**Yes**
Continuation on Main Thread	**Yes**	**Yes** (optional)	**Yes**	No
Continuation in Parallel	No	**Yes**	No (optional)	**Yes**
Set Degrees of Parallelism	No	No	No	**Yes**

Resources

Code Samples & Resources

https://github.com/jeremybytes/sdd-2023

Thank You!

Jeremy Clark

- www.jeremybytes.com
- jeremy@jeremybytes.com
- @jeremybytes

https://github.com/jeremybytes/sdd-2023