Tristan Izlar

COP2362

# TUTORIAL 7-1: Improving Throughput by Using Tasks

I worked alone.

Graphical user interface, text

Description automatically generated

Diagram

Description automatically generated

Could not get gaps/holes

MainPage.xaml.cs

using System;

using System.Diagnostics;

using System.IO;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Media.Imaging;

using System.Threading.Tasks;

using System.Threading;

namespace GraphDemo

{

public sealed partial class MainPage : Page

{

private int pixelWidth = 15000;

private int pixelHeight = 10000;

private WriteableBitmap graphBitmap = null;

private int bytesPerPixel = 4;

private byte[] data;

private byte redValue, greenValue, blueValue;

private CancellationTokenSource tokenSource = null;

public MainPage()

{

this.InitializeComponent();

int dataSize = bytesPerPixel \* pixelWidth \* pixelHeight;

data = new byte[dataSize];

graphBitmap = new WriteableBitmap(pixelWidth, pixelHeight);

}

private async void plotButton\_Click(object sender, RoutedEventArgs e)

{

Random rand = new Random();

redValue = (byte)rand.Next(0xFF);

greenValue = (byte)rand.Next(0xFF);

blueValue = (byte)rand.Next(0xFF);

tokenSource = new CancellationTokenSource();

CancellationToken token = tokenSource.Token;

Stopwatch watch = Stopwatch.StartNew();

Task first = Task.Run(() => generateGraphData(data, 0, pixelWidth / 4, token), token);

Task second = Task.Run(() => generateGraphData(data, pixelWidth / 4, pixelWidth / 2, token), token);

try

{

await first;

await second;

duration.Text = $"Duration (ms): {watch.ElapsedMilliseconds}";

}

catch (OperationCanceledException oce)

{

duration.Text = oce.Message;

}

string message = $"Status of tasks is {first.Status}, {second.Status}";

messages.Text = message;

Stream pixelStream = graphBitmap.PixelBuffer.AsStream();

pixelStream.Seek(0, SeekOrigin.Begin);

pixelStream.Write(data, 0, data.Length);

graphBitmap.Invalidate();

graphImage.Source = graphBitmap;

}

private void cancelButton\_Click(object sender, RoutedEventArgs e)

{

if (tokenSource != null)

{

tokenSource.Cancel();

}

}

private void generateGraphData(byte[] data, int partitionStart, int partitionEnd, CancellationToken token)

{

int a = pixelWidth / 2;

int b = a \* a;

int c = pixelHeight / 2;

for (int x = partitionStart; x < partitionEnd; x++)

{

int s = x \* x;

double p = Math.Sqrt(b - s);

for (double i = -p; i < p; i += 3)

{

token.ThrowIfCancellationRequested();

double r = Math.Sqrt(s + i \* i) / a;

double q = (r - 1) \* Math.Sin(24 \* r);

double y = i / 3 + (q \* c);

plotXY(data, (int)(-x + (pixelWidth / 2)), (int)(y + (pixelHeight / 2)));

plotXY(data, (int)(x + (pixelWidth / 2)), (int)(y + (pixelHeight / 2)));

}

}

}

private void plotXY(byte[] data, int x, int y)

{

int pixelIndex = (x + y \* pixelWidth) \* bytesPerPixel;

data[pixelIndex] = blueValue;

data[pixelIndex + 1] = greenValue;

data[pixelIndex + 2] = redValue;

data[pixelIndex + 3] = 0xBF;

}

}

}

App.xaml.cs

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices.WindowsRuntime;

using Windows.ApplicationModel;

using Windows.ApplicationModel.Activation;

using Windows.Foundation;

using Windows.Foundation.Collections;

using Windows.UI.Xaml;

using Windows.UI.Xaml.Controls;

using Windows.UI.Xaml.Controls.Primitives;

using Windows.UI.Xaml.Data;

using Windows.UI.Xaml.Input;

using Windows.UI.Xaml.Media;

using Windows.UI.Xaml.Navigation;

namespace GraphDemo

{

/// </summary>

sealed partial class App : Application

{

public App()

{

Microsoft.ApplicationInsights.WindowsAppInitializer.InitializeAsync(

Microsoft.ApplicationInsights.WindowsCollectors.Metadata |

Microsoft.ApplicationInsights.WindowsCollectors.Session);

this.InitializeComponent();

this.Suspending += OnSuspending;

}

protected override void OnLaunched(LaunchActivatedEventArgs e)

{

#if DEBUG

if (System.Diagnostics.Debugger.IsAttached)

{

this.DebugSettings.EnableFrameRateCounter = true;

}

#endif

Frame rootFrame = Window.Current.Content as Frame;

if (rootFrame == null)

{

rootFrame = new Frame();

rootFrame.NavigationFailed += OnNavigationFailed;

if (e.PreviousExecutionState == ApplicationExecutionState.Terminated)

{

}

Window.Current.Content = rootFrame;

}

if (rootFrame.Content == null)

{

rootFrame.Navigate(typeof(MainPage), e.Arguments);

}

Window.Current.Activate();

}

void OnNavigationFailed(object sender, NavigationFailedEventArgs e)

{

throw new Exception("Failed to load Page " + e.SourcePageType.FullName);

}

private void OnSuspending(object sender, SuspendingEventArgs e)

{

var deferral = e.SuspendingOperation.GetDeferral();

deferral.Complete();

}

}

}