Thumbnail Generation + Sim Theory's Thunder SDK: A Client's Case Study

Goal

In January 2025, Perception Grid[1] engaged Simulation Theory on a pilot project to enhance the performance of their spatial AI solution and demonstrate the power of the Thunder SDK.

The requirements of the pilot project were:

- ➔ A CPU-intensive computational load
- ➔ Open source
- ➔ Implemented in C#
- ➔ Create a demonstration application that Perception Grid can integrate into their technology suite that also works as a clear example of how to integrate and leverage the Thunder SDK

Magick.NET[2] was chosen because it meets the customer's requirements in the following ways:

- ➔ Magick.NET is reasonably CPU intensive
- ➔ Magick.NET is implemented using ImageMagick[3] (a commonly used package optimized internally using OpenMP)
- ➔ Magick.NET is open source with an Apache 2.0 license
- ➔ Magick.NET is implemented in C#

Hypothesis

Simulation Theory can accelerate Perception Grid's generation of thumbnails with a low-effort integration using the Thunder SDK and its C# bindings.

---

[1] https://www.perceptiongrid.io/
[2] https://github.com/dlemstra/Magick.NET
[3] ImageMagick is a powerful image manipulation library that supports over 100 major file formats (not including sub-formats). With Magick.NET you can use ImageMagick in your C#/VB.NET/.NET Core application without having to install ImageMagick on your server or desktop.

Results

A member of the Simulation Theory team completed an integration of the Thunder SDK into Magick.NET at the top level of the application. Post integration, on a system with 32 threads, the modified application reduced total execution time by 92.99% as compared to the original implementation.

Testing Criteria

The testing process involved resizing 400+ image files totaling 4.12GB. The files were each originally 4k by 4k pixels and were resized down to 200 by 200 pixels while maintaining the existing aspect ratio. The output format was png.

All testing was completed on an AMD Ryzen Threadripper 2990WX 32-Core Processor @ 3.0GHz, Windows 10 system with 128GB of RAM.

Original Application Execution time

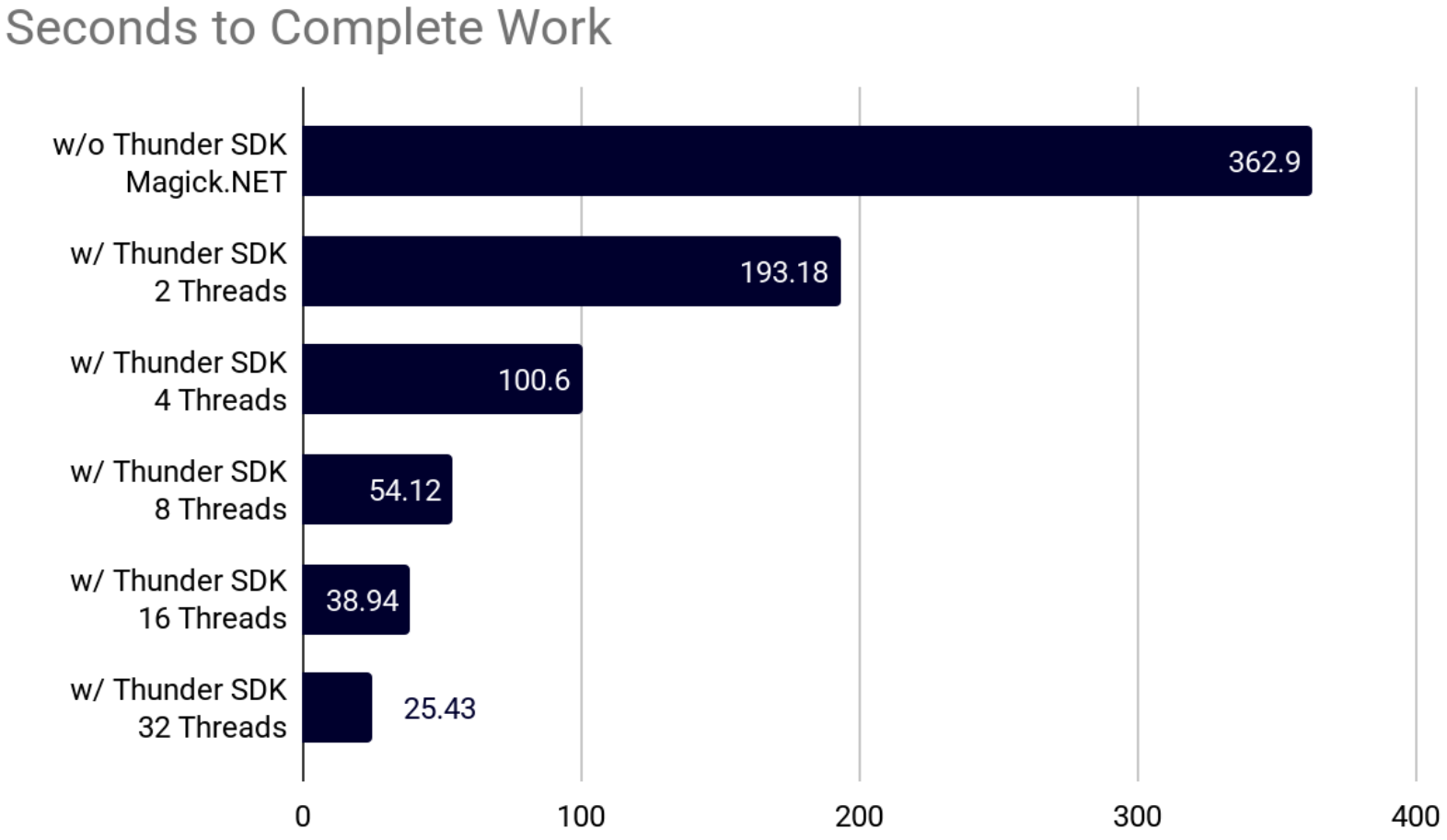
The original Magick.NET application completed all work in 362.9 seconds.

Modified Application Execution time

The Magick.NET application modified to use the Thunder SDK completed all work in 25.43 seconds, an improvement of 92.99% over the original application.

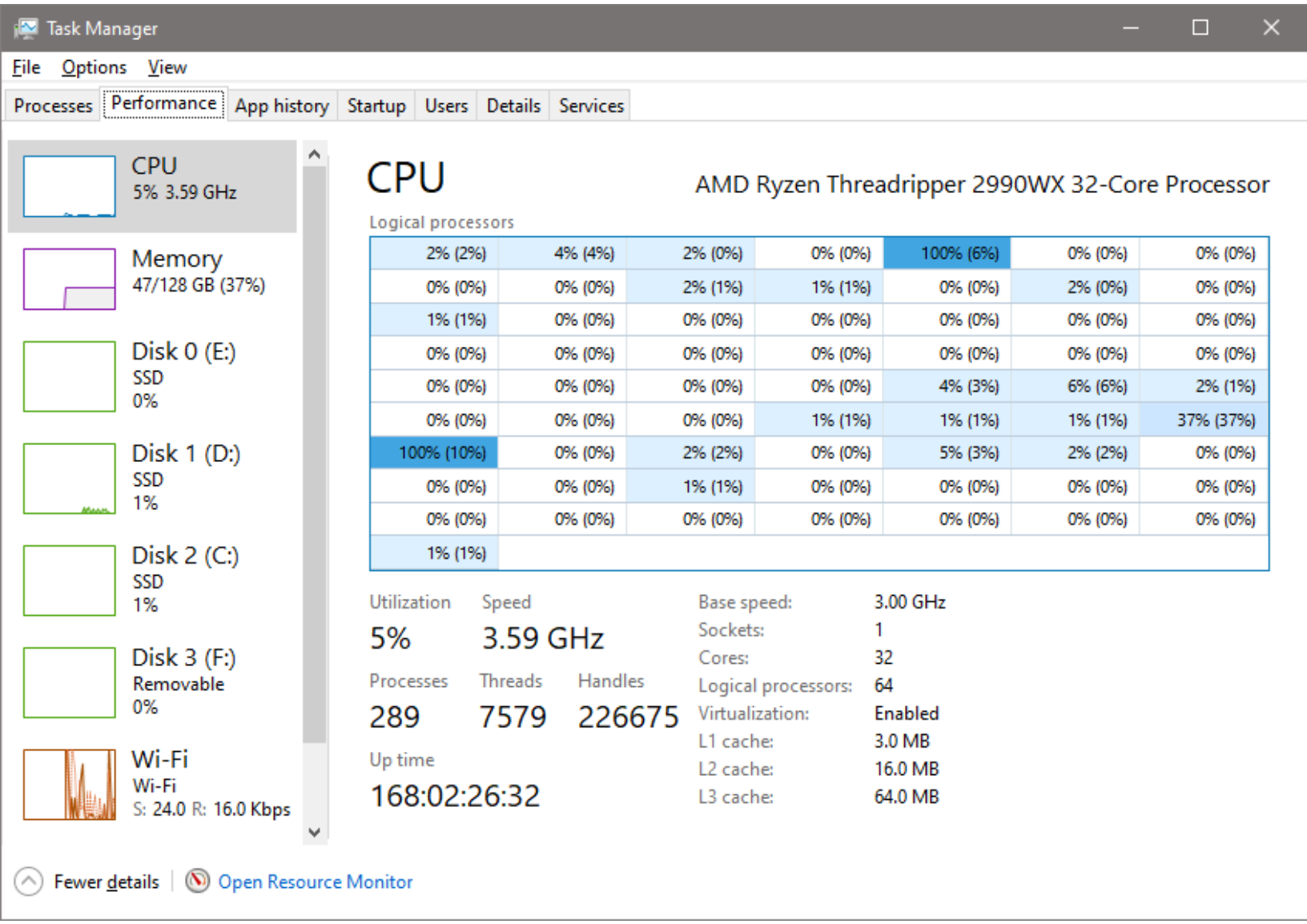
Using 2 threads, all work was completed in 193.18 seconds - reduced execution time by 46.76%
Using 4 threads, all work was completed in 100.60 seconds - reduced execution time by 72.27%
Using 8 threads, all work was completed in 54.12 seconds - reduced execution time by 85.08%
Using 16 threads, all work was completed in 38.94 seconds - reduced execution time by 89.27%
Using 32 threads, all work was completed in 25.43 seconds - reduced execution time by 92.99%

With further testing on more performance-optimized systems, we expect to see similar gains as thread count increases.


Seconds to Complete Work
w/o Thunder SDK Magick.NET
362.9
w/ Thunder SDK 2 Threads
193.18
w/ Thunder SDK 4 Threads
100.6
w/ Thunder SDK 8 Threads
54.12
w/ Thunder SDK 16 Threads
38.94
w/ Thunder SDK 32 Threads
25.43
0
100
200
300
400

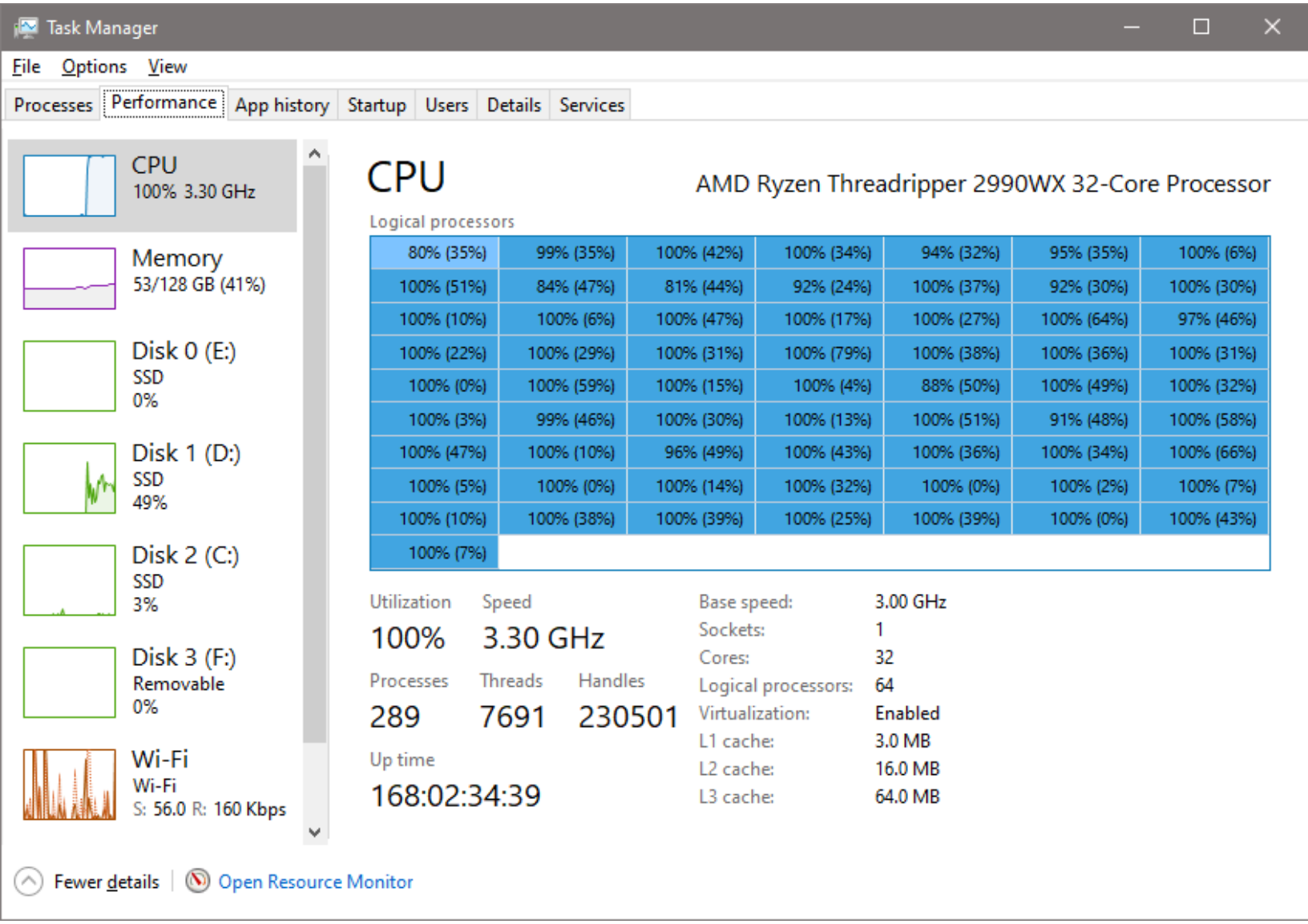
Original Magick.NET Application CPU Graph

The original Magick.NET application completed all work in 362.9 seconds.

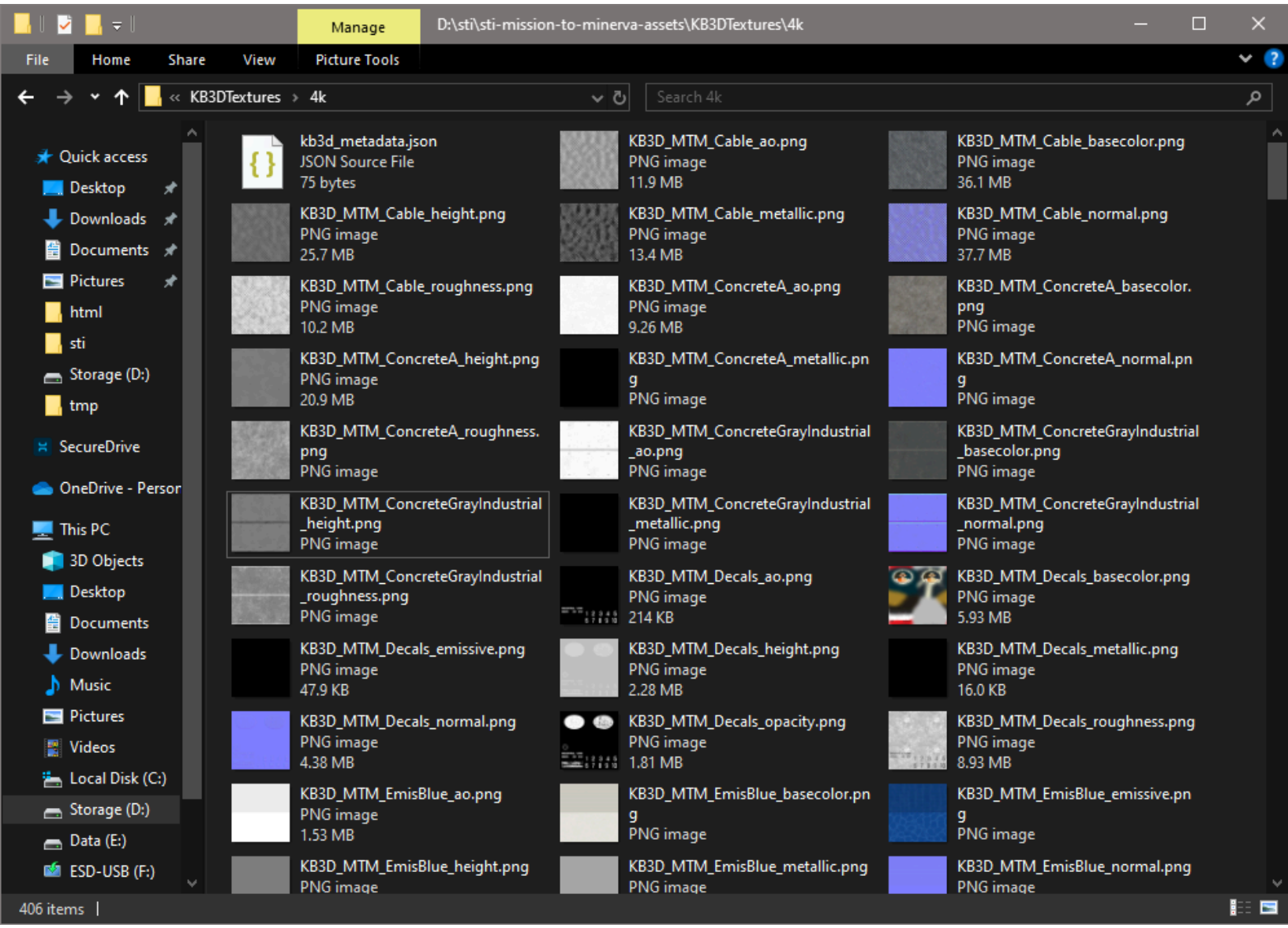


Magick.NET Modified with the Thunder SDK Using 32 Threads

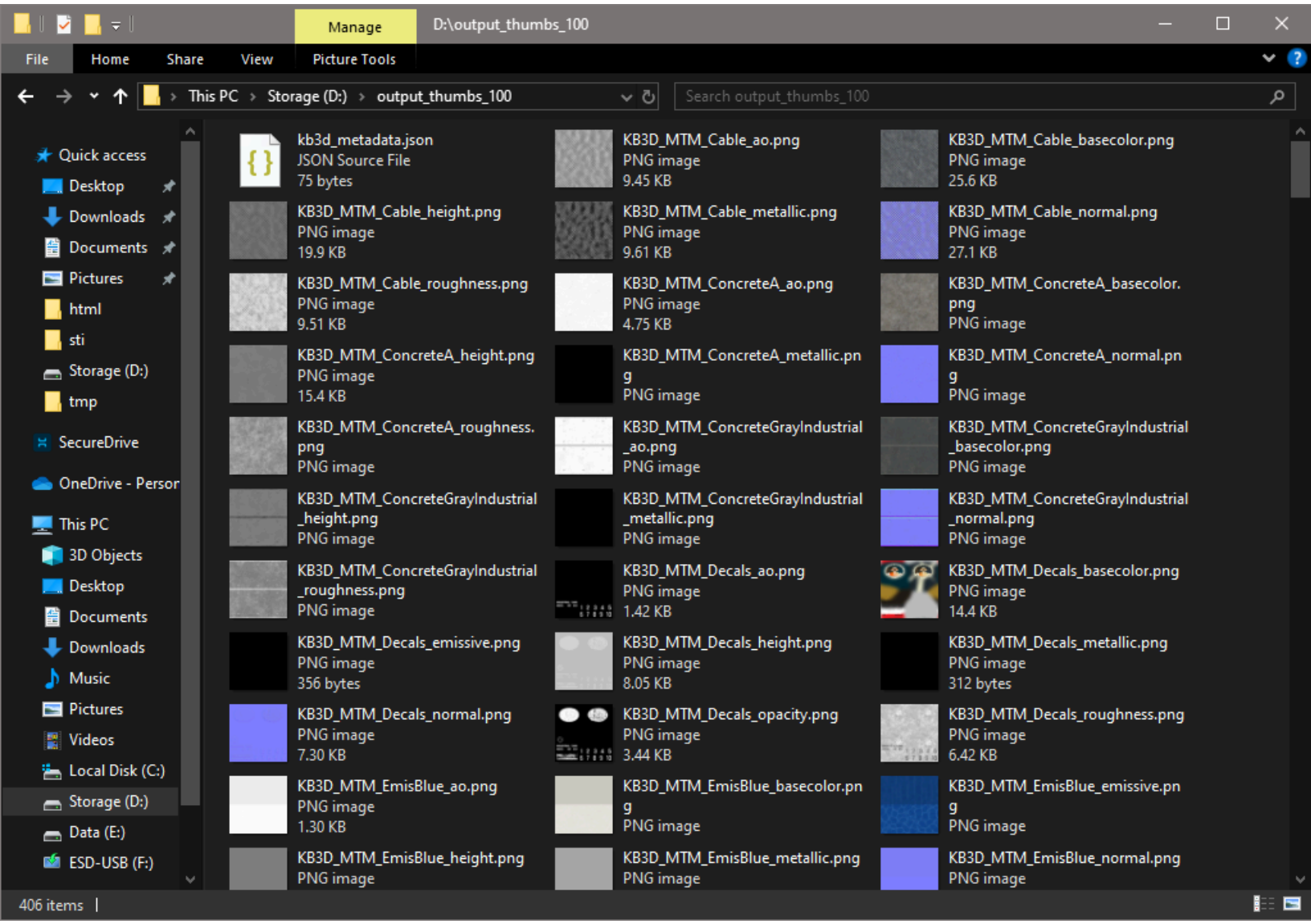
The work was completed in 25.43 seconds, reducing execution time by 92.99%



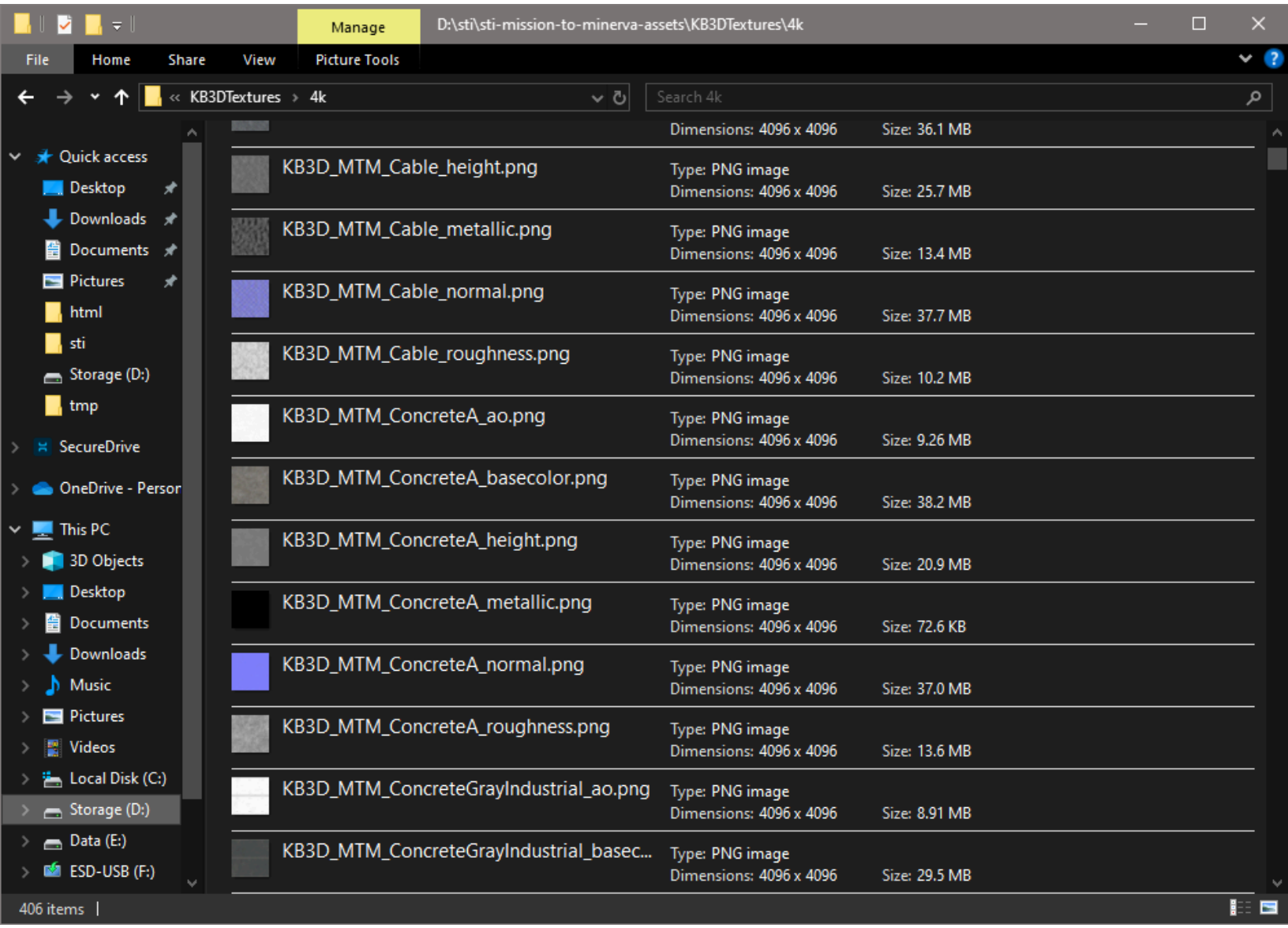
Thumbnail Source



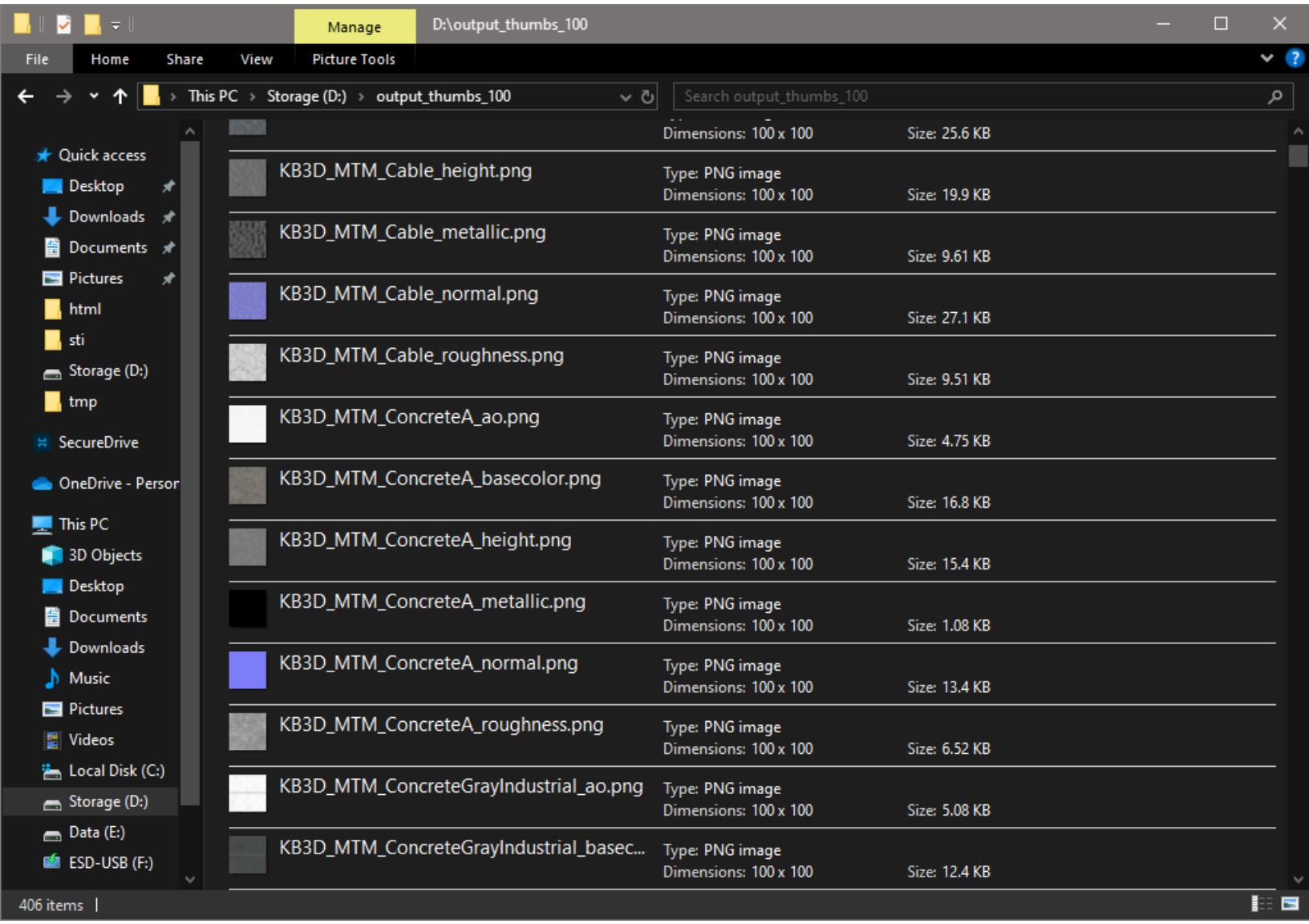
Thumbnail Destination



Thumbnail Source Content



Thumbnail Destination Content



Author: Randy Culley, CTO
sales@simtheoryinc.com
simtheoryinc.com