exacqVision Software and Multi-threading

Purpose

To make the exacqVision software experience more useful and efficient, exacqVision uses multi-threading to provide even better performance when used in combination with a hyper-threaded processor. This article defines multi-threading and hyper-threading, and then provides details about how exacqVision uses them.

What is multi-threading?

Multi-threading is a programming technique that allows a single process to open and use multiple threads. The threads are then able to execute independently and make the main process execute and respond faster, perform more efficiently, and reduce overall system usage.

What is hyper-threading?

Hyper-threading is a proprietary process developed by Intel that enables a multi-core processor to create two logical cores per physical core on the die. For example, a dual-core processor has two physical cores, but hyper-threading creates four logical cores for the system. If the operating system supports hyper-threading, it can then detect and use all four of these cores.

With multi-threading, what happens when you open the exacqVision Client software?

When the client software is opened, a single main thread is created for it. Additional threads are then opened based on the number of cores detected in the processor. These additional threads open and then wait to be used (for decoding frames from cameras). Additionally, two network threads are opened to process data coming in from servers. For example:

1. A non-hyper-threaded dual-core processor opens **five** threads: **one** main thread when the client opens, **two** decoding threads based on core count, and **two** network threads.

Created On: 11th February 2021
KB Number: KB-00302-302-210211

2. A hyper-threaded dual-core processor opens **seven** threads: **one** main thread when the client opens, **four** decoding threads based on core count, and **two** network threads.

What do these additional decoding threads accomplish for the client software?

The additional threads are responsible for decoding the frames displayed in the client software. Additional threads can be created by processes from a video card (such as Intel HD and NVIDIA).

How are these threads used to decode frames?

Frames come in from camera streams that are selected for display in the client software. The frames are decoded in the order in which they are received by the available threads. With a hyper-threaded quad core processor, **eight** threads are available to decode frames. After the frames are decoded by these threads, they are then displayed by a video component like Direct3D.

What happens if more streams are requested than there are available threads?

The additional streams are added in with the amount of decoding work done by the open threads. Load balancing is performed automatically, but the frames are decoded in the order in which they are received.

When do these additional threads close?

The additional threads do not close until the main thread closes, which occurs when the client software closes.

How are things different with the exacqVision Server software?

exacqVision Server is much simpler in that every installed plugin (such as Axis, psfpi, and archive) opens a thread. Like the client, these threads stay open while awaiting work. Because there is very little computation involved compared to the client, there might not be as dramatic of a spike in CPU usage when more data is incoming to the system.