Hardware Resource Guide

For Userful On-Premise

Revision History

Version	Date	Notes	Author
1.0	February 24, 2021	Initial Release	Greg Mitchell
1.1	May 28, 2021	Edited for Readability	Greg Mitchell

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

Introduction

Userful On-Premise servers are versatile, and their flexibility is unmatched by any other solution on the market. Because of this flexibility, you need to know the capabilities of your system during project planning.

There is no table or chart that gives a complete picture of these capabilities, nor is there a simple set of numbers that defines a server's capabilities. Depending on the content, a single server may support more than 100 screens, or the same server may be overwhelmed by multiple sources running on a single screen.

The simple answer to the question of what level and number of Userful servers you need for your project is to **talk to your Userful account manager or sales engineer**. Even after reading and understanding all of this information, it is highly recommended that you consult Userful before implementation, as software updates can often significantly improve functionality and limitations.

Certified Systems

The information here can be interpreted in many different ways and can be translated into all kinds of server and computer building ideas. If you plan to go your own way and order a Userful server from your own channels, this article is a great place to start.

This guide is limited to currently certified Userful systems.

GPUs

To date, Userful has used relatively inexpensive NVIDIA GTX consumer GPUs in conjunction with Zero Client devices. Since uClients are expected to gradually replace Zero Clients, this guide omits non-Quadro GPUs for planning and brevity.

Also, due to the wide variety of GPUs on the market, we will limit our discussion to the Quadro RTX 4000, 5000, and 6000, which are the certified models in Userful systems.

As of Userful 10.6, only one NVIDIA GPU can be used simultaneously in a Userful system, although there are plans for an architecture that will remove this limitation in the future.

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

Ingestion, Processing, and Output

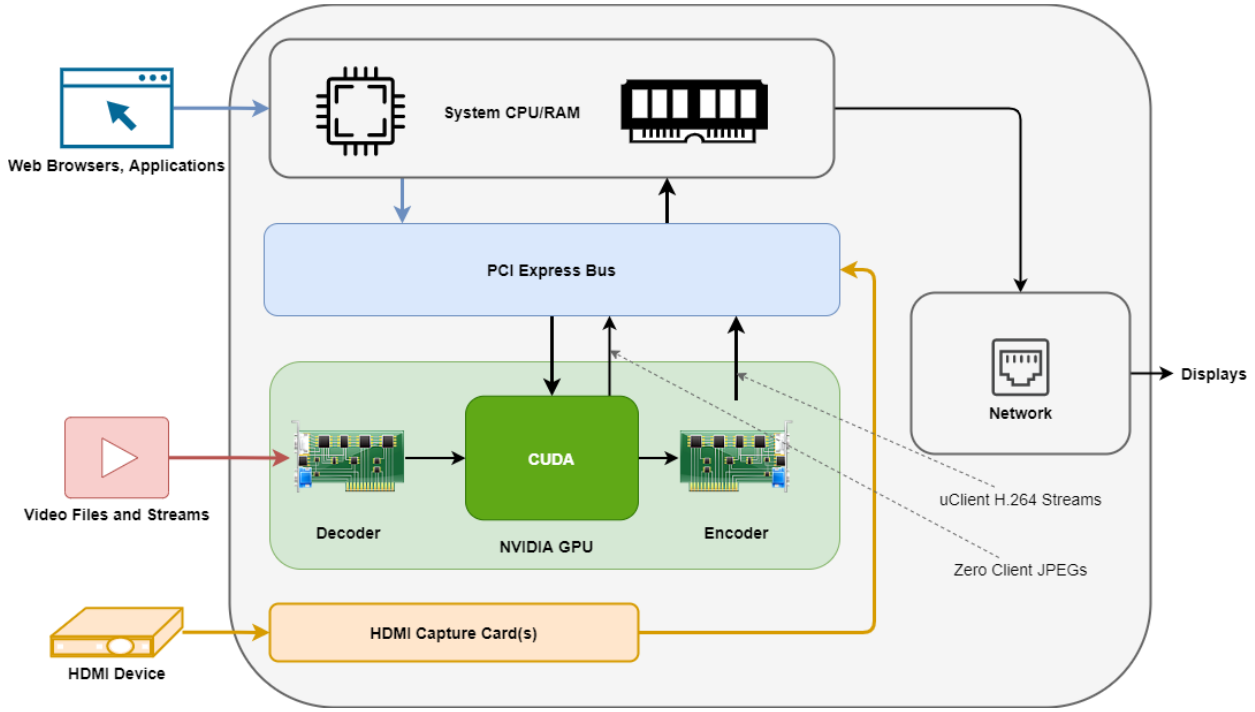
Resource usage must be considered in relation to the three main tasks that a Userful server must perform.

Ingesting content, either directly from a GPU decoder that receives and decodes video content, or from an application running on an interactive source, which must be sent to the GPU for processing. This is the most technically complex part of the process.

Processing means transforming and scaling the content from its raw state, or changing the data stream to fit the screen; Userful uses NVIDIA CUDA for this.

Encoding is the process of sending the transformed data to the display.

Data Flow Diagram



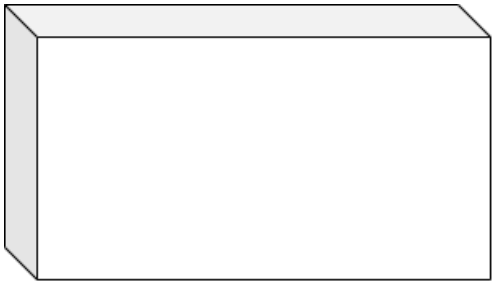
This is a simplified representation of the Userful server data flow. Content is taken from various parts of the system, passed through the PCIe bus, transformed by NVIDIA, and then encoded and sent to the endpoint. Each source type has its own path and different impact on resources.

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

Video Files and Streams, and the GPU Decoder

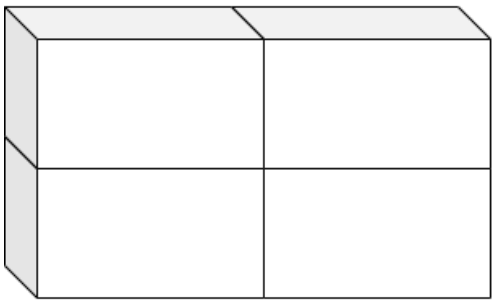
The direct source is a file or video stream. This is the media content that is decoded directly by the GPU decoder.

All supported GPU decoders can handle an 8K resolution workload at 60 frames per second. You can also think of this workload divided by resolution and frame rate as a canvas for the 8K60 limit. This is meant to illustrate the available bandwidth and is not limited to the example shown.



An 8K60 Canvas

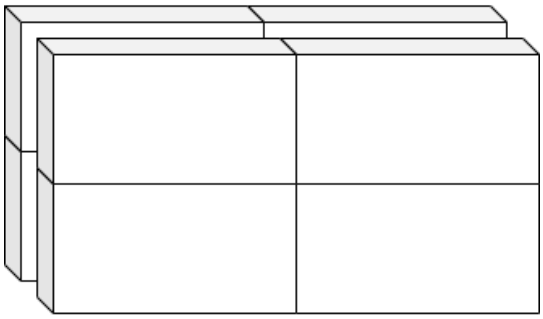
However, there are three very important limitations to this model that should be noted. The first limitation is that Userful is configured by default to only use hardware decoding for four direct sources at once. So even if your only direct source is five 1080p30 streams, the fifth will be routed to the CPU instead of the GPU. This can be changed by a Userful engineer in a configuration file on the system.



A 4x 4K60 Canvas

The second limitation is load, and the simple reality is that if you design a system to run at 100% capacity 24/7, there is no room for flexibility or growth.

The third limitation is file formats; there are only a limited number of formats supported by NVIDIA's hardware decoder. This information can be found here.

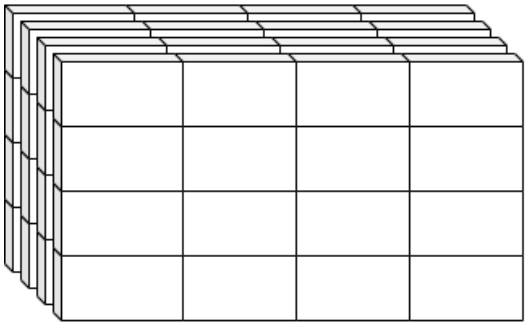


An 8x 4K30 Canvas

So, of all the major components, the main direct source limitation is the GPU decoder. However, with Userful's architecture, you can see that it is very flexible as long as you don't exceed the GPU's capabilities.

Overload Conditions

If the above limit is exceeded, Userful will attempt to decode all remaining streams with software decoding. This is a tactful way of saying that it will use the CPU to decode video. This is undesirable because we want to reserve the CPU for system operations and interactive sources. In addition, software decoding cannot match the quality or latency of hardware decoding; overloading CPU resources can make the system unresponsive and potentially unstable. It is also less efficient than GPU decoding.



A 128x 1080p15 Canvas

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

Exceptions

Blackmagic HDMI Capture sources do not need to be decoded. They are passed directly from the Blackmagic Capture driver to the GPU.

Using Forward and Store with Signage Player eliminates the need for live decoding. However, this only applies to Signage Player, and this delivery model is fundamentally incompatible with live streaming content.

Web Browsers and Applications

Interactive sources are instances of applications that run directly from the Userful server itself. They are powered by the server's operating system and do not consume decoder resources.

Interactive sources consume CPU, RAM, and PCI Express bandwidth.

CPU and RAM are simple: Userful servers are equipped with powerful i7 and Xeon processors and 32GB or 64GB of user-expandable RAM to handle multiple instances of most applications (web browsers, VNC and RDP clients) under normal circumstances. The exact number of these sources is largely limited by the applications running on it; a web browser displaying KPIs uses far fewer CPU cycles than a browser playing a YouTube video. For this reason, we do not recommend using a web browser or any other interactive source to play the video. GPU acceleration is not available, as it is for web browsers running on desktop PCs. The only exception is "Session Acceleration", which can only accelerate one session at a time.

There is no fixed or inherent limit to the number of interactive sources that can run on a Userful server. What you do need to be aware of are the limitations of PCI Express and the encoder.

PCI Express Upload to the GPU

Frames from the interactive source are sent to the GPU for processing and encoding. The amount of data passing over the PCI Express bus is calculated using the following formula:

Width*Height*bpp*FPS/8

- Width: Resolution width
- Height: Resolution height
- bpp: Bits Per Pixel (colour depth)
- FPS: Frames Per Second
- /8: 8 bits to a Byte

As an example, if we use a Web Browser source, running at 4K 30FPS:

- Width=3840
- Height=2160
- Web Browser bpp=32
- FPS=30
- /8

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

This gives a required bandwidth of **3840*2160*32*30/8=995328000**, or about **1GB/s**. In other words, for a Userful system with current specifications (as of February 2021), the PCIe 3.0 bandwidth of a GPU running at full 16x bandwidth would be able to support about 12 of these sources. In future systems with PCI Express 4.0 and 5.0 motherboards and GPUs, this limitation will increase.

Different Sources vary in their colour depth:

Source	Description	Colour space	BPP
Ximage	All Interactive sources, HDMI capture with Xclient enabled	BGRx	32
HDMI capture	All Blackmagic Capture sources (Xclient disabled)	UYVY	16
Software decode	Direct sources that exceed the hardware decoder limit	I420	12

Here is a handy fact sheet with the most commonly used and recommended values for interactive sources.

Use Case	Resolution	Framerate	BPP	Bandwidth
1080 Web Browser, VNC, RDP, Desktop	1920×1080	30	32	250MB/s
Cable TV Capture	1920×1080	59.94	16	250MB/s
4K HDMI Capture	3840×2160	60	16	1GB/s

At this point, you may be wondering why we haven't talked about these sources along with the direct sources and decoders. This is because the work that the GPU decoder does is already in the GPU, so it doesn't need to be constantly streaming over the PCI-Express interface. As such, it is not applicable in this case. The bandwidth used to load media resources onto the GPU itself is negligible compared to system operations.

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

CUDA Processing

When content is decoded or loaded, it temporarily resides in the GPU's VRAM. It doesn't take much, but this is where the real magic of Userful happens. We use the NVIDIA CUDA platform to run a series of tasks on your content. These are in order of execution:

Color Transformation - All content is converted to the BGRA color space.

Compositing - If you're using multi-window, picture-in-picture or Command & Control, this is where all the stitching, scaling and compositing happens. These are the most computationally intensive operations.

Cropping and scaling - If you are using a zero client, this is where the content is scaled up or down to fit the required workspace.

JPEG encoding - If you're using a zero client, the content is directly encoded and loaded as JPEGs.

These tasks are very difficult to specify in writing in terms of the amount of CUDA compute engine resources they consume. Each application has a different impact on these metrics. This is the main difference in performance between the three cards, which is also the reason why higher levels of Userful servers have been equipped with higher-performance GPUs with a proportionally increasing number of CUDA cores (all of the listed cards have identical encoding/decoding capabilities and PCI Express buses).

Card	CUDA Cores	Server
Quadro RTX 4000	2,304	Standard
Quadro RTX 5000	3,072	Professional+
Quadro RTX 6000	4,608	Enterprise and others

You may have noticed that the Quadro RTX 8000 is not included in this list. This GPU has a significant increase in VRAM over the RTX 6000, but other than that it is the same for CUDA cores and encoding/decoding capabilities. Real-time operations don't use as much VRAM.

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

Encoding

Once the content has been downloaded and processed, the next step is to send it to the client and display. There are two ways to do this, depending on the type of client you are using.

Zero Clients

The zero client receives the content as a JPEG file. Converting the content to a JPEG file is not computationally expensive, so this step is done in the CUDA processing stage. All that remains is to send the data from the GPU to the zero client driver and then to the zero client via the network card.

The network bandwidth available for this process is highly dependent on the content being transferred: video traffic throughput at 60 frames per second for a single zero client can exceed 100 Mbps. On the other hand, static images and slideshows that are updated only a few times per minute generate little traffic. That's why the Zero Client is equipped with a Gigabit Ethernet port, and why the network design requirements for the Zero Client are so stringent.

In addition, Zero Clients have more stringent latency requirements than uClients because they communicate with the server in both directions.

uClients

uClients are a more traditional video endpoint: they receive traffic via a dedicated RTSP stream that carries video in H.264 or H.265 format. The creation of this stream is done by a GPU encoder.

The GPU encoder has the same functionality as the decoder, so it has the same "canvas" options as the ingest measurement. What is different now is that all sources are eligible for this 8K60 canvas, including direct video, web browsers, and HDMI captures. Each video stream that is sent will consume a portion of this encoder.

Why highlight the streams? Because it is not per screen. For example, if you are playing a 4K60 video on a 3×3 video wall, it will only consume the resources needed to play a 4K60 video, not a 5K60 video. The scaling of this operation is done on the uClient side. And this means that it is theoretically possible to power three identical walls, or play the same 4K60 video on a 4×4 or 5×5 wall.

As another example, playing a 1080p60 video on a single display consumes an expected amount of encoders. However, playing the same video on 16 Mirror Group displays would fill up encoder resources because each stream would need to be generated independently. We are working on a solution to this issue for a future release, but no specific release date has been set.

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.

Transmission

When the content is ready to be transmitted, it will be sent as a standard RTSP video stream.

The bandwidth of this stream depends on the options set in Zone Settings.

When "Optimized for Video" is selected, Zero Client's JPEG files are compressed to 87% (more compression, less detail), and 4:2:0 color. In this configuration, the Zero Client can generate up to 128 Mbps of traffic.

When “Optimized for Text” is set, the JPEG file is compressed to 92% (less compression, more detail) and at full 4:4:4 color space. With this configuration, the Zero Client can generate up to 480 Mbps of traffic (not recommended unless used with very low FPS applications).

Note that the above configuration does not apply to uClients. uClients will generate about 12Mbps of traffic for a 1080p60 video stream and up to 40Mbps for a 4K60, so use the settings to monitor your network bandwidth wisely. The “Optimize for Video/Text” settings do not affect uClients.

Tips for Managing Resources

There are several ways to optimize Userful server performance.

Use Monitoring to track system resource usage and establish a baseline. This is necessary to see what is happening on the server at any given time, and if any of the various components - CPU, RAM, GPU, PCI Express, network, etc - are reaching capacity.

By default, Userful 's server limits video encoding to 30FPS. You can manually enable 60FPS in Performance Settings in the Control Center. Please note that this will double the load on the encoding side, so at least check Monitoring before enabling this option.

The Web Browser source has a built-in frame rate control feature. For dashboards and metrics that are updated very infrequently, you can set the FPS down to 0.1, which can save a lot of resources.

Conclusion

So far, we've covered the basics of how Userful On-Premise captures, processes, and distributes content. These tips are intended to help Userful partners and customers plan a successful implementation.

We strongly recommend consulting with a Userful account manager or sales engineer when planning your implementation, and as a "second set of eyes" our experience and expertise is always available to our customers.

Copyright © 2021 Userful Corporation. All rights reserved. All other trademarks are the property of their respective owners.