CueTip

Tips for GeoCue Users

Increasing Data Production Efficiency by Deploying Low-Cost Distributed Processing

For information on this CueTip™, contact:
Martin Flood
GeoCue Corporation
9668 Madison Blvd., Suite 101
Madison, AL 35758
mflood@geocue.com
01-519-590-8291

Purpose:

This technical note is a companion to the briefing note *"Command Dispatch System in GeoCue Enterprise"*, which describes GeoCue's Command Dispatch System in detail and includes technical information on the architecture and implementation of distributed processing functions within GeoCue. The purpose of this companion note is to discuss how distributed processing can be deployed to improve overall efficiency for a specific workflow – in this case lidar data production.

In the context of geospatial data production, there are three valuable, but difficult to implement features that can directly impact the efficiency of a particular production environment:

- ✓ Running commands on a computer other than the one from which the command is invoked.
- ✓ Scheduling a command to run at a later time.
- ✓ Distributing commands that operate on multiple objects across multiple computers.

Beyond an obvious extension of technical capabilities on the production floor, we firmly believe introducing distributed processing moves geospatial production from 'workstation' to 'enterprise' in ways that can have tremendous positive impacts on individual and team productivity and, ultimately, job profit.

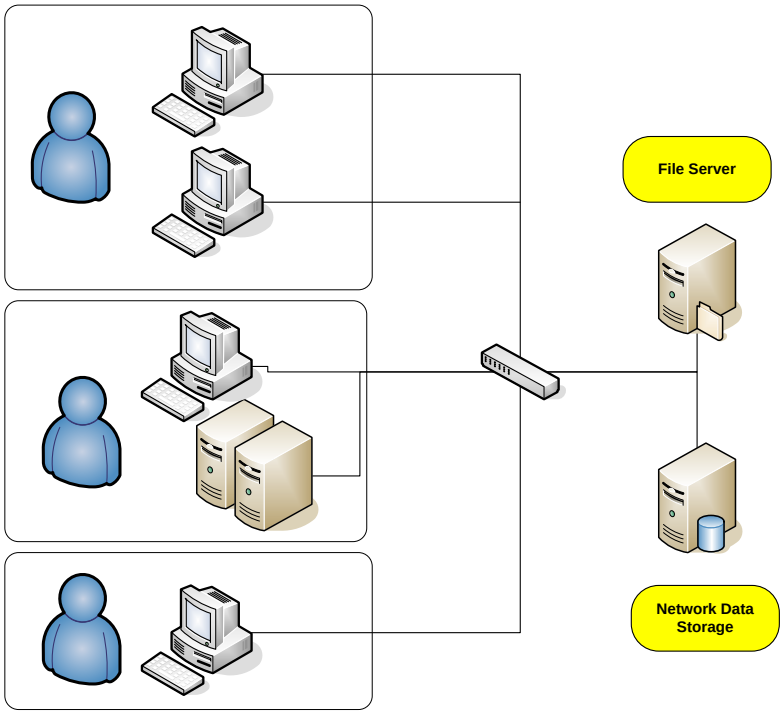
These three features have all been included in V3.0 of our GeoCue software. Collectively they are referred to as the Command Dispatch System (CDS). The CDS is included with any 'Enterprise' (floating) license of GeoCue. It is not included with the 'Workstation' (node-locked) version. The three major features added to GeoCue by the CDS are:

Remoting or remote execution - running processes on a machine other than the one on which it is initiated. This is valuable in a number of circumstances. One obvious case is when a user would like to set up a task on their workstation via the interactive GUI, but would like for the computationally intensive part of the command – the 'number crunching' part – to run on a different machine, freeing up her workstation for the next task. An example might be to set-up an import of lidar flight lines into a project using a client workstation, but instruct the command to run on a remote server.

Scheduling commands extends the value of remote execution by allowing a user to specify a date and time when the command is to start. An example might be to set-up the macros that are to classify a ground surface from lidar data, but specify that the actual processing is not to start until everyone has gone home for the evening, thus preventing overloading workstations when other tasks need to be done.

Distributing tasks that can be split up across multiple computers is the final key concept needed to extend *remoting* and *scheduling* to a true 'distributed processing' environment. A classic example of distributed processing is to spread the rectification of 100 images across 10 computers and do this from a single GUI interface on a project management machine. A similar example in a lidar production environment would be spreading 100 ground classification macros across 5 separate machines.

1. To provide an example of the benefits of deploying distributed processing, we will consider a typical lidar data production set-up. We will limit the discussion to the necessary processing from delivery of the field data through to product delivery. A typical production infrastructure set-up to support this particular workflow can be represented as follows:



   Each member of the production team has a dedicated workstation (or workstations) through which they interact with the data – that is complete their assigned production tasks – using a variety of software tools. The geospatial data itself generally resides on networked data storage or is copied to the local machine to reduce network bandwidth problems. Both automated tasks (e.g. macros) and manual tasks (e.g. interactive editing) generally run locally on the user's machine or on a 'remote' machine that is physically close enough that the user can kick-off processes on the machine and then return to his own workstation.

2. The efficiency of this production set-up, essentially the amount of data that can flow through the production floor in a given time, is limited by basic factors such as:

a. The number of people. The more staff, the more effort available in a given period to complete tasks requiring user input (editing) or to manage automated 'batch' processes (macros).

b. The number of computers. The more CPUs available, the more automated batch processing tasks, such as macro processing, can be completed in parallel.

c. The speed of the computers. The faster the CPUs, the faster automated tasks can be completed.

d. Interactive editing time. The amount of effort required to manually edit and clean-up the data after automated processing is a key component of overall production efficiency, production cost and schedule time.

e. Network bandwidth performance. The faster data can be moved around the network, when required, the more flexible the infrastructure.

2. For lidar data production the number of people required – staffing of the production floor – is often dominated by the need for final manual editing, clean-up and QA/QC of the data. The more manual work need, the more people are required to complete the work in a given time period and the higher the costs. This is the main reason we see so much emphasis across the industry on outsourcing manual editing to lower-cost production shops and developing more efficient automated classification routines. Deploying distributed processing capabilities does not directly address this particular factor, interactive editing will still be required regardless of how any automated batch processing is completed. However, using distributed processing to assign all automated tasks to remote nodes, in a controlled and efficient manner, does free-up local machine resources and user time that would otherwise be spent monitoring 'batch' processes. It allows each user to make more effective use of their own time with a corresponding increase in their personal efficiency (more manual editing completed in a given time, since less time is spent monitoring automated processes).

3. Given these constraints, our objective from a production management viewpoint is to optimize each of the above factors while maximizing throughput. We want to generate the most revenue for the lowest cost in a fixed time period, while maintaining the necessary professional and quality standards. It helps that items (a) and (b) in our list are actually linked. It is of very limited benefit to add more production staff if they don't have a computer to work with and conversely, adding more machines without an efficient method for the existing staff to integrate them into the production workflow usually results in idle machines. Adding one without the other does not improve your production efficiency (although both will increase your costs). Note that while it is usually the case that adding more people requires you to add more computers, there are exceptions. The most obvious is a production shop that runs multiple shifts with people 'sharing' computers to complete their interactive tasks such as data editing. In a shift environment, making sure the 'local' machines are free at the beginning of each shift for interactive editing – not running an automated batch process – becomes even more critical to maintaining production efficiency.

4. Often, what is practically important in a real-world production shop is not the absolute number of staff or computers, but the ratio of people to machines; the number of CPUs each person can realistically control to handle automated tasks, while working interactively on their own workstation (or in some extreme cases we have encountered, interactively working on several workstations at the same time). This span of control is typically no more than 2-4 machines, and control is usually accomplished by some combination of remote access

software (remote desktop) to access the other machines from another location (e.g. 'home'), co-locating multiple computers in a single cubicle and 'hot-seating' or physically moving from workstation to workstation. We have seen production shops that employ all three of these methods in an attempt to improve their throughput, usually with limited success. While each of these 'distributed processing by user' methods has some advantages, each also has serious drawbacks and limitations. Primarily they suffer from very limited scalability. It is very inefficient, using these methods, for a single person to control 10 or 20 nodes, especially if each machine needs to be interacted with through a different GUI (even if the GUI is actually accessed remotely from the user's own workstation). Even harder for them to efficiently allocate work across these machines based on CPU load or available resources at the moment a particular task is dispatched. Other limitations of this 'distributed processing by user' approach include:

a. There is no central management or control of all 'distributed' process across all machines available on the network. So, while one user can manually distribute and monitor work across the various 'local' machines under her direct control, there is no coordination with other users doing the same thing in other areas of the network.

b. Automated batch processing of macros, using GeoCue to assign and run a macro on multiple tiles or by using a TerraScan project structure, is possible. However without true distributed processing, these approaches load the local CPU where the process is running, reducing its availability for any other tasks for the duration of the batch process.

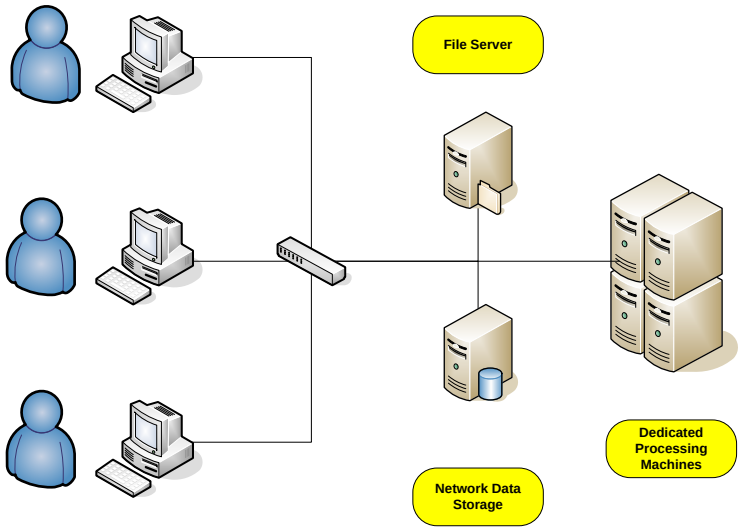
c. Failures during stand-alone batch processing can result in an abrupt conclusion or corruption of the batch process. For example, if tile #23 of a batch of 250 tiles causes the local machine to hang, there is no gentle error trapping and recovery that allows the unprocessed tiles to continue on another, different machine.

d. The user has no ability to automatically take advantage of idle machines, reconfigure the batch process 'on-the-fly' as new machines become available, or selectively burden power machines (workhorses) with the bulk of any batch processing effort.

e. There is no inherent ability, beyond what a particular software tool may provide, to leverage multiple core, multiple processor machines for added advantage.

The above limitations can be removed by deploying a true distributed processing environment that includes integrated *remoting*, *scheduling* and *distributing* capabilities, all controllable from a single GUI and visible to all users across the network. These functions are all part of the Command Dispatch System (CDS) in GeoCue V3.0. With the CDS, it is relatively straightforward to enable desktop distributed processing for your existing data production workflow.

5. To take advantage of GeoCue's Command Dispatch System – to deploy true desktop distributed processing capabilities – requires only a very simple reconfiguration of the production infrastructure outlined above.



Each user now needs only a single dedicated workstation, primarily for performing any interactive tasks, such as manual editing or setting-up automated batch processes, while the bulk of any automated tasks are dispatched to an array of remote processing nodes. While this seems like a relatively minor modification to the infrastructure, there are several major advantages to deploying GeoCue's distributed processing in the configuration shown above:

a. You can quickly and easily deploy additional remote processing nodes for your production floor and control them through GeoCue's CDS.[i]

b. Central control and management of all machines on your network, both local machines and remote processing nodes, is available through GeoCue's Dispatch Monitor.

c. The GeoCue Dispatch Monitor automatically handles intelligent queuing and processing of all the dispatched/distributed tasks (so users never have to check to see which remote nodes are 'free').

d. Intelligent load balancing will ensure that the bulk of the work will be done by the faster/more powerful machines on your network, resulting in shorter overall processing times.

e. Error trapping and recovery is much more robust than in other remote processing scenarios since a single process failure – a single tile crashing or single node going offline – will not stop the entire batch process.

f. Scalability for the majority of automated tasks – for instance to increase the number of macros run from 1,000 tiles/month to 10,000 tiles/month – is now achieved by adding more low-cost computers, not more staff.

g. Individual processes – for example each tile in a block of tiles undergoing macro processing – completes separately and the tile is immediately ready for further processing. Users do not have to wait for an entire batch process to finish before starting the next production step.

h. Any user, with the appropriate permissions, can now queue up automated task for remote processing on the first available machine.

i. The integrated scheduling capabilities allow users full control of when particular batches run (such as at 3:00 in the morning when CPU load is at a minimum).

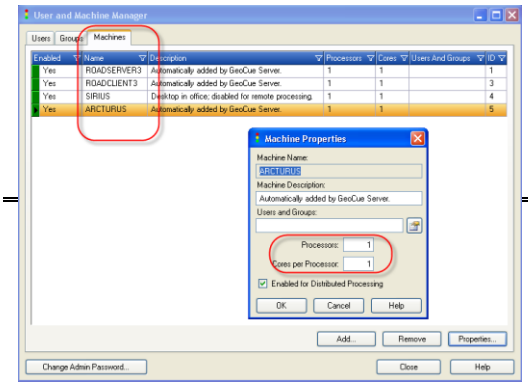
j. Distributed processing capabilities can be easily extended to other software tools beside TerraScan – as long as the process is automated (needs no user input) – or even to your own proprietary code, using GeoCue's Environment Builder

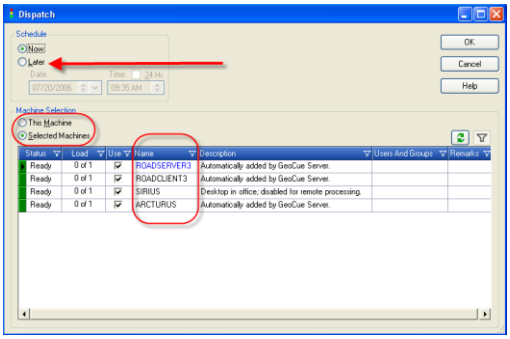
6. To further expand on the distributed processing concept for lidar data production, we are working with Terrasolid to integrate a new TerraScan 'slave' processor into the command dispatch system. The slave module will not require a full TerraScan license or any MicroStation license; the requirements will be a GeoCue remoting module as well as the Terrasolid slave module, significantly reducing the software license cost/node. This new software approach will make processing lidar data using arrays of slave computers an extremely cost-effective way to significantly improve production throughput.

7. Finally, there is also a compelling 'single seat' argument for using distributed processing in a lidar data production environment, even without deploying an array of remote nodes as discussed above. GeoCue's CDS includes the ability to process dispatched TerraScan macros in a MicroStation 'silent' mode. 'Silent' mode runs MicroStation as a background Windows Service without any GUI or interactive dialogs. This frees resources for any interactive – GUI-based – editing being done on the local machine. For example, a single user with a single machine can 'dispatch' a batch of macros to their own machine to process as a background service and as soon as the first tile completes they can bring it up in TerraScan on the same machine for QA/QC and interactive editing.

Conclusion:

We believe distributed processing represents a major leap forward in efficiency for geospatial data production, allowing you to significantly improve the efficiency of your existing resources and to mange larger and more complex projects by adding cheap, dedicated computer resources rather than adding more staff. Being able to improve your data production efficiency and scale-up your production operations – to do more, in less time, with less (or the same) resources – is becoming critical to maintaining a firm's competitiveness, especially as prices continue to drop, data volumes increase and clients demand shorter and shorter schedules. We see many firms struggling with this issue across our industry; with lidar data production in particular. Given the cheap cost of computing power, we feel gains in efficiency and production throughput can be achieved more cost-effectively by deploying tools that leverage adding cheap computing power over adding more people; tools that maximize the use of your existing infrastructure while minimizing the number of people needed to manage each CPU to its full capacity, all while maximizing the total number of CPUs each person can control. GeoCue' new distributed processing capabilities clearly fall into this category.





If you have any difficulties or questions in implementing this CueTip, please do not hesitate to contact me or one of the other GeoCue staff.

[i] This is a very cost-effective approach to increasing your current production team's efficiency and improving your production throughput. Currently, a dual-processor, dual-core blade server, 3.0 GHz with 8 GB RAM is available for under $2,500 US. An array of four such servers can be deployed for under $10,000 and, using the CDS, allows a single user to distribute all automated processes, such as running TerraScan macros (eventually without the need for MicroStation licenses) or intensity image generation, across 16 processor/cores from a single GUI with just a few mouse clicks. This approach becomes particularly efficient when you need to process 100s or 1000s of working tiles in a short time period.