Row64: How GPU Enabled Spreadsheets Are Unleashing The Power Of Big Data For The Everyday Analyst

Direct-to-hardware programming and low-latency optimization are delivering 1000x the speed of traditional spreadsheets with the enhanced functionality of big data systems.

Spreadsheets are arguably the software that saved the personal computer. Introduced first to the world with Visicalc, the utility of spreadsheets was so immediate and profound that Steve Jobs singled them out as the single largest propellant for Apple's early success.

Since then spreadsheet dominance has continued, with Excel topping the charts for most popular PC program for the last 5 years. However, the functionality of Excel has begun to hit its limits in recent years, as its legacy code simply cannot keep up with the size of modern data sets that span from 30,000 rows to more than 1 million. Often analysts fighting against large data sets have to *turn off spreadsheet calculations entirely* just so their computers can load them. The so-called "spreadsheet lag" is a known industry truth and it's estimated that data analysts currently spend 80% of their time simply prepping data.

The result is billions of dollars lost to productivity and opportunity cost. A recent report by Forrester estimated if data visibility were to increase even 10% for a typical Fortune 1000 company, revenue would increase by roughly $65 million. This number is only expected to increase as mobile devices, IoT, genetic sequencing, ecommerce and financial products are both generating new data and automating at exponential rates.

Companies with the adequate resources and personnel have tried to tackle their big data problems by either turning to relational database management systems (typically querying in SQL) or by using intermediary software such as ETL programs (Extract, Transform, Load) to ingest and clean data before porting it to Excel in manageable quantities.

The problem is all of these systems are still imperfect, and often costly. Analysis software such as SAS/STAT can cost upwards of $430 a month per user. Data dashboard services such Tableau can cost into the thousands , and still requires analysts to learn how to query in SQL. PowerBI, while cheaper, runs into constant bottlenecks when its data warehouses are continuously queried by analysts.

Together, this barriers to entry ahve meant big data analysis has largely been in the realm of large companies (500+ employees) locking out a majority of small and medium sized businesses that could hugely benefit from understanding their data. In fact it's estimated that 73% of data is left unused for analytics—an opportunity cost estimated to be in the trillions of dollars.

Furthermore, existing software isn't properly optimized for modern computing power, meaning up to $28 billions of dollars a year spent on hardware improvements is largely wasted, as software is leaving idle the most powerful processing units of the computer. Companies are spending large amounts of capital to get a fractional return on their speed—simply because their programs aren't optimizing their extra processing power.

Row64 has thus emerged as the first true solution to bridge the costly mismatch between big data needs and simple, yet powerful analytics solutions. By rethinking how data software is coded from the ground up, Row64 combines the power and flexibility of big data systems with the ease and familiarity of Excel—all at record-breaking speeds. In total, Row64 brings the power of big data science to the everyday analyst.

To learn about how Row64 is able to achieve its record breaking speed, scale and simplicity, we take a deep dive under the hood to explain the innovative technology behind our GPU spreadsheets.

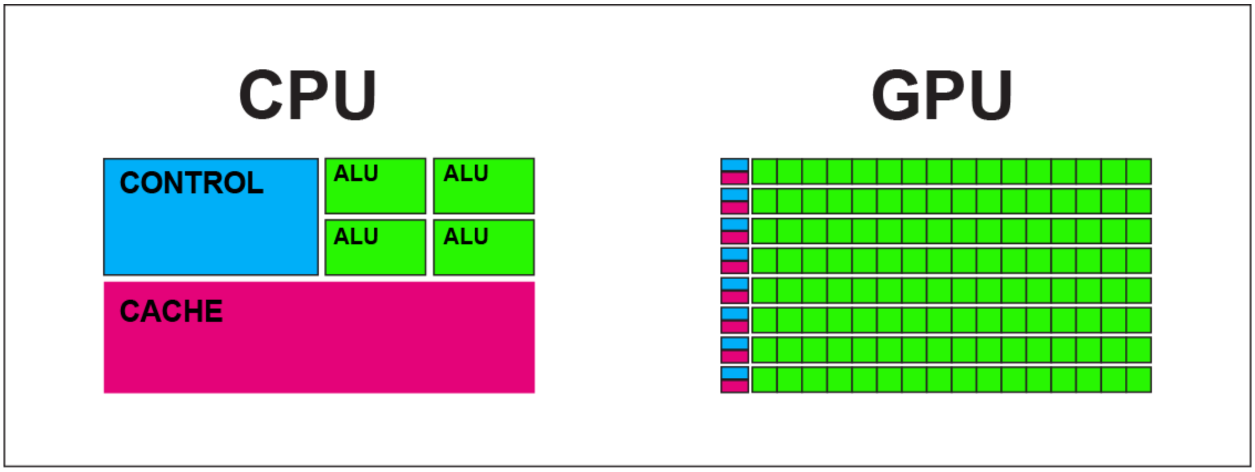
GPU Programming: Optimizing Software For Hardware

Often when we think of the most impressive computing capabilities on personal computers, we think of games. Their intensely-detailed worlds are both expansive and hyper quick—enabling a user to react responsively in real time with a constantly changing virtual environment.

Compare that with the user experience with Excel—a flat white sheet displaying simple ASCII symbols—and it's hard to understand why the latter has load speeds that at times are reduced to an utter crawl.

The reason for this painstaking slowness is *a lack of GPU optimization*. Simply put, the code behind all spreadsheet software is not properly leveraging the full value of their machine's processors.

Games—which require thousands of polygons to be rendered every second—are written to take advantage of computer GPUs, which are highly specialized to handle heavy mathematical computations. GPUs process large amounts of data in parallel, achieving one specific output in a very short time.

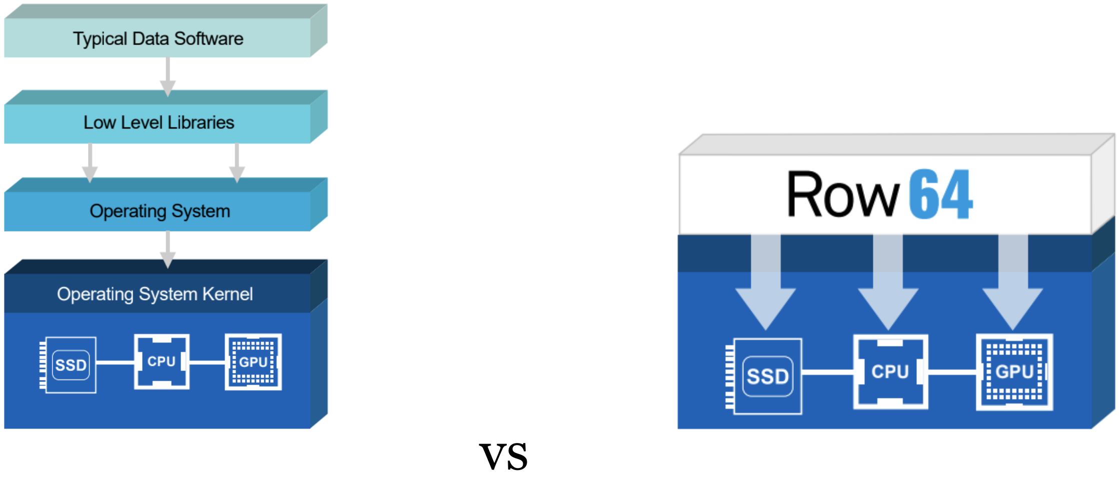


Architectures of GPU are designed for highly parallel functionality, with multiple identical rows of control units, processing units (ALUs) cache memory working in tandem for large computational tasks.

CPUs on the other hand are optimized to "think", following a path where they load instructions from memory, decode them, and then execute their respective function. A GPU would be terrible at processing the variety of tasks a CPU has to balance, and CPUs could never approach the speed of GPU for highly parallelized computations.

So why are spreadsheets (which are in essence giant graphic calculators) not using GPUs? The answer is because writing software to GPUs requires coding at a low level, i.e. programming directly to the hardware. For many software companies, this is extremely costly, as the process is highly time consuming (requiring a lot of stability checks and iterations) and there are way fewer programmers available who specialize in low-level GPU programming. As a result, legacy companies continue to instead iterate on code that was written for CPUs way before the incredible advancements in NVIDIA and AMD graphics cards were available.

By programming to the convenience of the coder, *not* the machine, spreadsheets are underutilizing powerful resources— forcing tons of data through unspecialized CPUs—while leaving largely untouched the incredible computational power of GPUs.



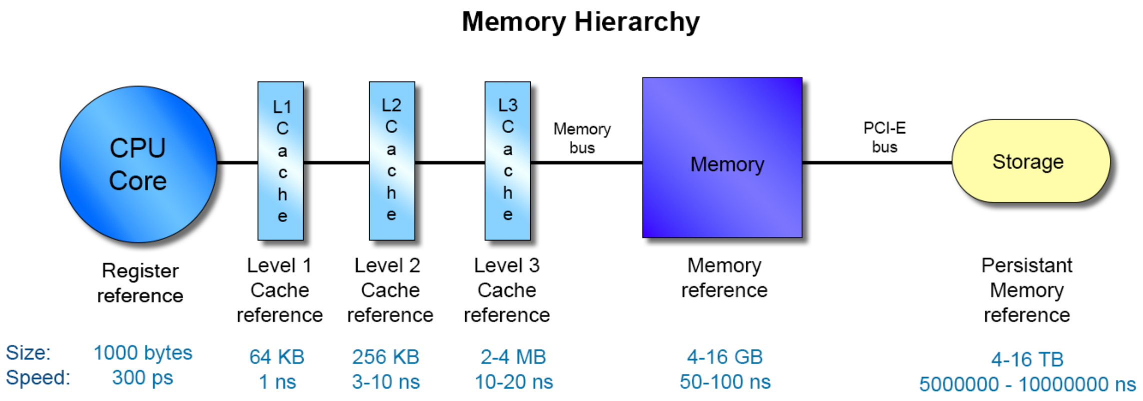
By programming direct to the hardware via "bare-metal" programming, Row64 eliminates the decision trees and abstraction of high level programming and smartly allocates directly to memory, CPU and GPU

Row64 solves this by programming everything *directly* to the hardware. This type of "bare-metal" programming takes full advantage of the processing power of the GPU, while maintaining the same easy UI for the end user. The result is that our GPU-enabled spreadsheets (or GPU Spreadsheets), are able to achieve speeds up to 1000x faster than traditional spreadsheets using the *same exact computers.*

Customized Computing: Smart Loads, Hybrid Engines and Low Latency Optimization

The first step of Row64 was to create a program that enabled GPU computations to be used in spreadsheet calculations. However, we soon realized that even *faster* calculations were possible by optimizing beyond the GPU, to *every* component of a computer—from the SSD down to the cache.

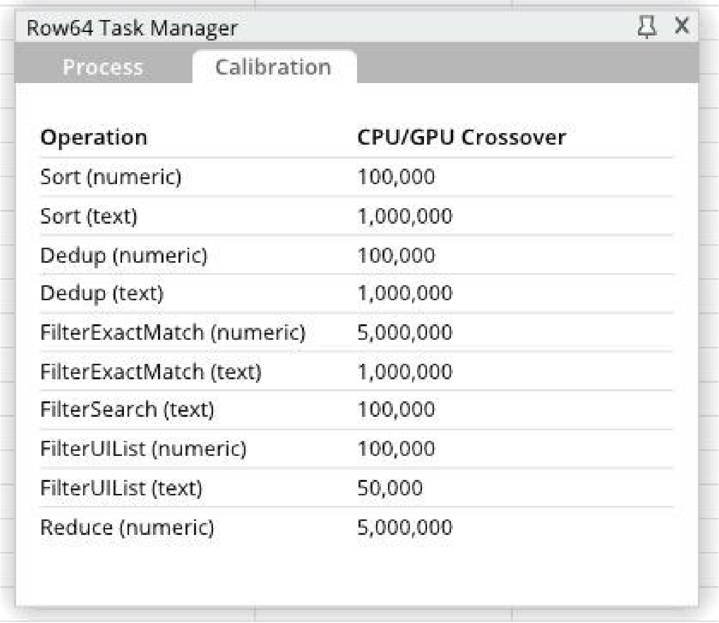
A big reason this is improved speed is the concept of *low latency*. Those familiar with computer architecture are aware of the blazing fast speeds of the cache, from the fastest level of L1 down to L3. This is largely due to the location of the cache itself, which is stored on the shortest possible path to the CPU (often located on or right next to the CPU itself).



The location of memory is a crucial component of speed. Cache memory is the fastest of all types of memory often sitting adjacent to or on top of the CPU itself.

Similarly the location of data on a computer's storage system can be optimized for improvements in speed by reducing the physical distance the data needs to travel to get to the processors (referred to "short read" optimization). This is possible both on SSDs and physical hard disks, via "short stroking".

Due to the principle of low latency, processes on the CPU will be faster than processes on the GPU *if* the CPU can handle the process. This means for low load tasks, the CPU may *actually* be better suited than the GPU. We dub this the "CPU/GPU crossover threshold". Tasks below this threshold are allocated to the CPU for short distance reads, and tasks above this load level are demanding enough to require the GPU—even if it means increasing the distance needed for the computation. This is the core essence of our secret, the "***hybrid engine"*** approach.



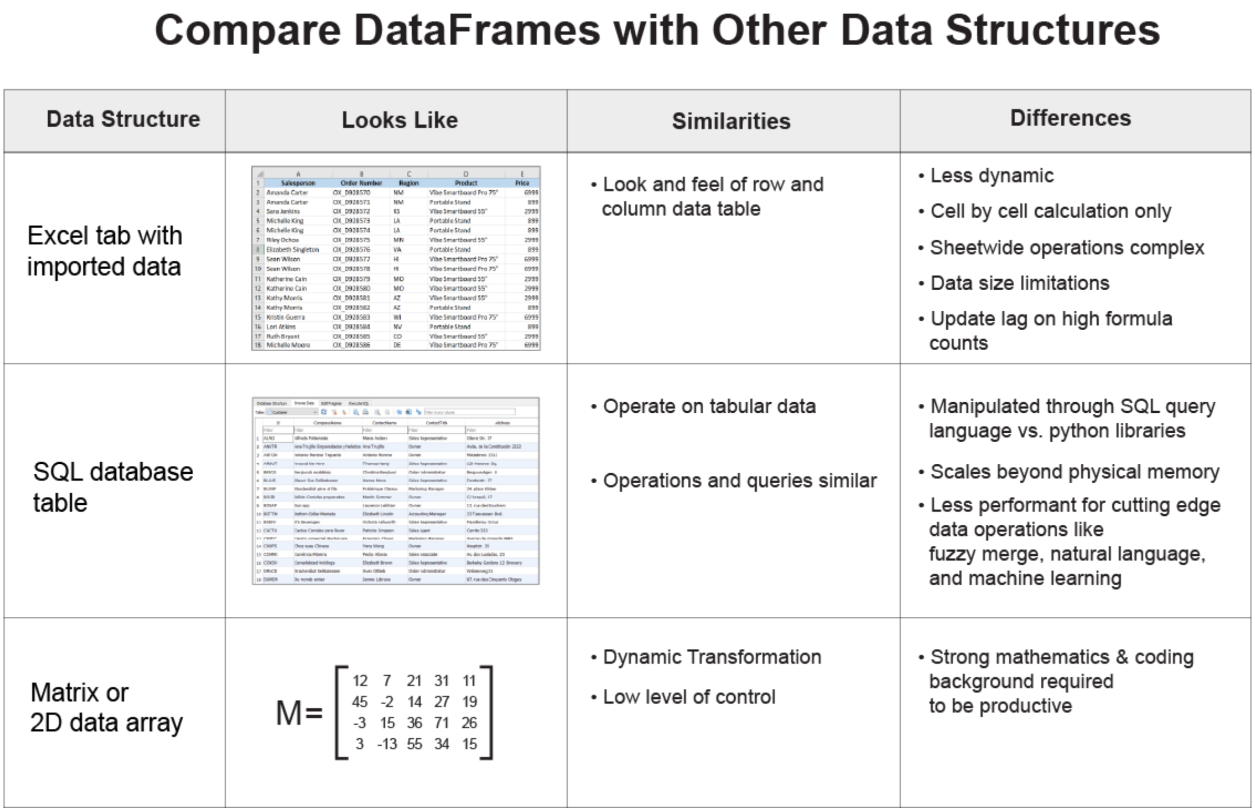
Operation	CPU/GPU Crossover
Sort (numeric)	100,000
Sort (text)	1,000,000
Dedup (numeric)	100,000
Dedup (text)	1,000,000
FilterExactMatch (numeric)	5,000,000
FilterExactMatch (text)	1,000,000
FilterSearch (text)	100,000
FilterUIList (numeric)	100,000
FilterUIList (text)	50,000
Reduce (numeric)	5,000,000

A look at the CPU/GPU crossover threshold shows how our hybrid engine allocates tasks to processors both by type of function and load size

To take full advantage of all the components at your computer's disposal and deploy our hybrid engine technology, we developed a truly unique "smart load" system, that runs a full analysis on your computer at the hardware level, and optimizes to each specific component. This ensures that we are pushing every computer to its limit, maximizing your specific copy of Row64 to your specific device—in stark contrast to the one-code-fits-all model of all legacy spreadsheet software.

Bringing The Power Of Python To Excel With DataFrames

A key component for how Row64 is able to deliver record breaking speed and functionality is the optimized use of *DataFrames*. We like to think of DataFrames as the foundation for spreadsheets with superpowers, because they deliver Excel-like simplicity with enhanced flexibility and Dataframes are structured similarly to spreadsheets—with rows and columns—however, instead of needing to be manipulated as a flat file, cell-by-cell, they can be manipulated as entire sheets or rows. This means that rather than doing 100 individual calculations for a 2 column x 50 row spreadsheet, a DataFrame can do just two calculations, and then express each cell as part of its respective column or row.



Compare DataFrames with Other Data Structures

Data Structure	Looks Like	Similarities	Differences
Excel tab with imported data	A B C D E 1 Salesperson | Order Number | Region | Product | Price 2 Amanda Carter | OX_0928570 | NM | Vibe Smartboard Pro 75" | 6999 3 Amanda Carter | OX_0928571 | NM | Portable Stand | 899 4 Sara Jenkins | OX_0928572 | KS | Vibe Smartboard 55" | 2999 5 Michelle King | OX_0928573 | LA | Portable Stand | 899 6 Michelle King | OX_0928574 | LA | Portable Stand | 899 7 Riley Ochoa | OX_0928575 | MN | Vibe Smartboard 55" | 2999 8 Elizabeth Singleton | OX_0928576 | VA | Portable Stand | 899 9 Sean Wilson | OX_0928577 | HI | Vibe Smartboard Pro 75" | 6999 10 Sean Wilson | OX_0928578 | HI | Vibe Smartboard Pro 75" | 6999 11 Katherine Cain | OX_0928579 | MO | Vibe Smartboard 55" | 2999 12 Katherine Cain | OX_0928580 | MO | Vibe Smartboard 55" | 2999 13 Kathy Morris | OX_0928581 | AZ | Vibe Smartboard 55" | 2999 14 Kathy Morris | OX_0928582 | AZ | Portable Stand | 899 15 Kristin Guerra | OX_0928583 | WI | Vibe Smartboard Pro 75" | 6999 16 Lori Atkins | OX_0928584 | NV | Portable Stand | 899 17 Ruth Bryant | OX_0928585 | CO | Vibe Smartboard 55" | 2999 18 Michelle Moore | OX_0928586 | DE | Vibe Smartboard Pro 75" | 6999	• Look and feel of row and column data table	• Less dynamic • Cell by cell calculation only • Sheetwide operations complex • Data size limitations • Update lag on high formula counts
SQL database table		• Operate on tabular data • Operations and queries similar	• Manipulated through SQL query language vs. python libraries • Scales beyond physical memory • Less performant for cutting edge data operations like fuzzy merge, natural language, and machine learning
Matrix or 2D data array	$M = \begin{bmatrix} 12 & 7 & 21 & 31 & 11 \\ 45 & -2 & 14 & 27 & 19 \\ -3 & 15 & 36 & 71 & 26 \\ 3 & -13 & 55 & 34 & 15 \end{bmatrix}$	• Dynamic Transformation • Low level of control	• Strong mathematics & coding background required to be productive

Similarities and differences between the DataFrames, Excel tabs, database tables, and a 2D Matrices .

In recent years, Dataframes have become popular among heavy data users (most notably in the Pandas Python library) for their incredible speed and flexibility. The problem is that for non-coding business analysts, programming in Python or R can be daunting and prohibitive.

By creating a simple and familiar UI wrapping in a DataFrame architecture, Row64 brings the speed and power of Python based DataFrames, to a no-code Excel-style spreadsheet—with all the benefits of GPU enabled calculations outlined above.

Financial Implications of Row64's Increased Speed

The level of speed and scale unlocked by Row64 isn't just a marginal upgrade, it's a step-change function increase. By enabling analysts to tailor their workflow around data insights rather than *data limitations*—Row64 has the power to fundamentally change how companies interact with their data.

To understand the magnitude of this potential change, it's worth understanding that the world's treasuries—the highest level of financial institutions on the planet—are wasting roughly 625 work days a year *just waiting for Excel to load tasks as simple as checking their balance.*

This is a tremendous waste of highly skilled labor hours. Business analysts are spending hours compiling ledger data for monthly revenue summaries when they could be doing it instantaneously. The ability to be unencumbered by data means summaries and schedules could be created daily or even weekly with little to no time loss from the analyst side. Accounting consultants—who often charge $400/hour and higher for their services—could see an entire shift in their business model, once their time spent cleaning data for Excel gets reduced to zero. In effect, virtually all analyst and consultant time could be spent on decision intelligence (not load time).

The ability to scale data instantaneously doesn't just mean an improvement on current practices—it actually introduces entirely new revenue streams entirely. An example of this is when for the first time ever, Row64 was able to price the *entire equity options market in real time,* a feat which shifts the framework around the possibilities of high frequency trading.

Similarly in the hard sciences, gene sequencing has been shown to be cut from 1 ½ days ***to just 20 minutes*** when using the power of GPU computations. This means world altering outbreaks like Covid can be sequenced and new variants identified in minutes instead of days—potentially preventing new world outbreaks.

Beyond calculations, the ability to do search and drill down functions has enabled security agencies like the NSA to immediately parse through security logs as big as half a terabyte, finding crucial information like hacker IP addresses in just a couple seconds.

The limitless nature of GPU enabled data analysis in essence isn't a new model of a car, but the actual creation of a modern combustion engine itself. We believe that with this new data toolset, Row64 could help propel us to a new era of responsive data-based, decision intelligence.

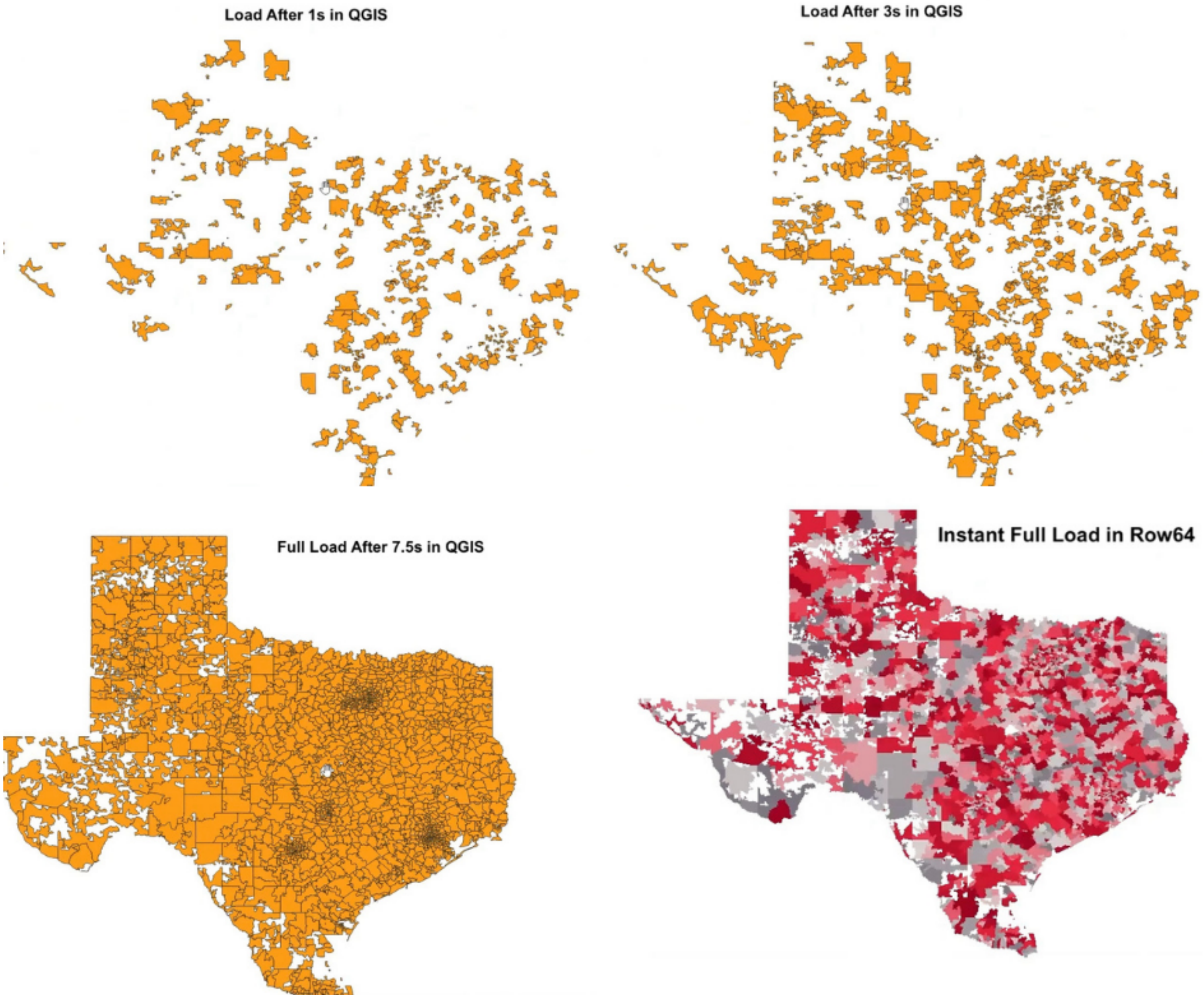
Seeing Is Understanding: How Row64's Next Generation Data Visualizations Translate Speed Into Intelligence

The lasting impact of the printing press wasn't the speed with which it helped ink hit paper, but the speed with which those printed words distributed new information to the masses.

Similarly, the power of Row64 doesn't come just from the speed with which it can sort and analyze data, *but the insight which that data analysis provides.* By doing away with the time needed to prepare data, we are opening up a new frontier of hyper-responsive, real-time data visualization tools, showcasing everything from real time geo-plotting, to animated time series and bar chart racing.

These visualizations are highly efficient at communicating information—so much so that a study by the Wharton School of Business found that data visualization could shorten meetings by up to 24%. With visual information processed roughly 60,000 times faster than text, effective data visualization tools can increase understanding of everything from communicable disease spread to trends in ecommerce.

By harnessing the same hybrid engine from our GPU Spreadsheets, Row64 is able to deliver data visualizations that operate at a speed and scale not possible with popular data visualization tools such as Plotly, Power BI or Tableau.



This figure shows identical files that map all 1,930 zip codes in Texas. Row64 is able to zoom in and out in realtime, where QGIS takes up to 7.5 seconds each time the user wants to scroll in or out.

By creating a simple to use program that can quickly ingest, sort and then visualize data, Row64 is delivering a truly end-to-end upgrade for every facet of the modern analyst's needs.

Row64 Speed and Scale Records

The best way to understand just how fast Row64 is to directly compare the blazing fast speed to the competition. We outline a few of the most relevant benchmarks below.

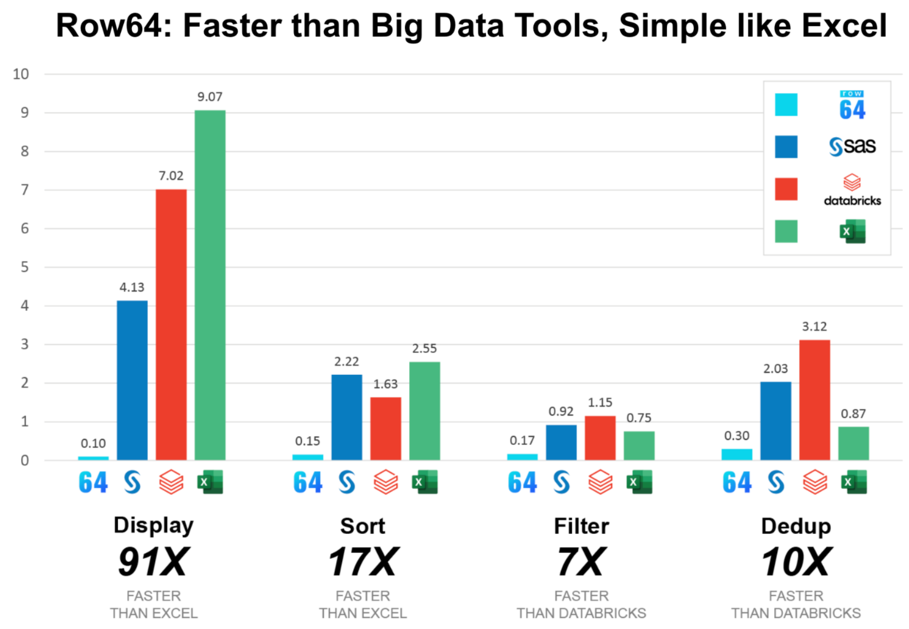
- **Opening Files (12x faster).** Row64 can open a 108 million line flat file in 10.4 seconds[1]. In Anatella, opening the same sized files takes ***2 minutes 3 seconds.*** Microsoft Excel cannot even load file sizes this big (cap is 1 million rows).

- **Sorting (17x faster)** . Excel takes 2.55 seconds to sort a 900k row spreadsheet. Row64 did the same computation in **0.15 seconds[2]**.

- **Filter (7x faster).** Sorting the same 900k row file, Row64 was able to complete the task in 0.17 seconds, 7x faster than DataBricks.

- **Real Time Equity Market Pricing (1154x faster).** Row64 was able to price 1 million Options from the equity market in 0.26 seconds[3]. This is ***1154x times faster*** than the industry target of *5 minutes.*

- **1 Billion Load Scale (First Ever).** While Excel has a 1 million row limit, Row64 can open a record breaking 1 ***billion*** rows, the first spreadsheet software that can handle this load scale. We've tested load and scroll functionality at this scale on computers as old as the 2013 ThinkPad.



Speed comparison of 900K record file (Excel cannot load large datasets). Time in seconds. Faster speed is better. Tested on Intel i7 (6 Core), Nvidia GeForce RTX 2080, 16 GB RAM.
Row64 compared with SAS 9.4, Apache Spark 3.0.0, Microsoft Office 365, Excel Build 13029.20308

The level of speed and scale unlocked by Row64 is beyond a marginal upgrade. It's a stepchange function that could fundamentally change paradigms in how analysts must shape their workflow around data. By handing over supercomputer-like ability to everyday data analysts, we hope to unlock a new frontier in big data and software.

[1] Calculations done on a 2021 RTX 3080 PC.

[2] Tested on Intel i7 (6 Core), Nvidia GeForce RTX 2080, 16 GB RAM. Microsoft Office 365, Excel Build 13029.20308

[3] Tested on AMD RX6900 XT Dual GPU.