Vectorization

The New Era of Big Data Parallelism

Every five to 10 years, an engineering breakthrough emerges that disrupts database software for the better. In the early 1990s, MPP parallel server clusters emerged, becoming widespread in the 2000s. Later, in-memory databases accelerated response times by eliminating the need to access disks. Recently, cloud elasticity and tiered storage improved costs and reduced risk. Vectorization is the newest breakthrough gaining momentum towards widespread adoption. Early adopters are using fully vectorized databases to foster new applications and reap lower costs. To best understand where vectorization is going, let's first understand where we've been.

WHITE PAPER

The Speed of Need

Since the first electronic computers, computer speed has been an obsession. Every year we buy faster CPUs, faster memory, faster storage, and faster software. Historically, the primary buying criteria for database software is also performance, performance, and more performance. We are all impatient to finish our tasks. But why does computer performance matter so much?

Foremost, we equate system performance with efficiency. Our personal efficiency is tied to the computer's speed. Sitting and waiting for the computer can be boring, often frustrating. The faster the computer runs, the more likely we can deliver results to management on time. Poor system or software performance means getting stuck waiting late into the night or working weekends. Faster performance also means:

- **Accuracy:** Initial results in every analytic report have errors or rough edges. The analyst adjusts parameters, adds data, and runs it again. The analyst does this many times until the accuracy improves. Which often leads to better customer retention, better pricing, or cost savings.
- **Scalability:** It's no secret that corporations are inundated with data nowadays. Good scalability means that workloads that took 18 hours to run years ago can now get done in under a minute. Impossible workloads from last year become hourly tasks today. Tackling impossible workloads with scalability is often a competitive advantage.
- **Real-time:** Some business problems can't wait. It's better to stop fraud now, not after the money disappears. Supply chains schedule trucks and laborers hour-by-hour. GPS delivery rerouting needs to be done today.
- **Complex math:** Many algorithms consume extreme amounts of CPU capacity. Car manufacturers running car crash computer simulations use massive server farms which is hundreds of times cheaper than actually crashing cars. Security experts analyzing web pages to detect hackers strain huge server farms too. But the paybacks are huge.

Parallel database software is the heart of analytic workload performance. Parallel databases enable new applications, new insights, and better employee productivity. In this paper we explore advances in database parallel processing and their foundations.

As the table below shows, hardware designs have been perfected for decades. Some of these technologies take many years of incubation before reaching widespread adoption. Consequently, most database software uses only SMP servers. MPPs are clusters of SMP servers. It's more difficult to build software for an MPP cluster. The SIMD hardware (GPUs and vector cores) are installed in the SMP and MPP servers. That means four kinds of parallelism are now available for database software to exploit.

Hardware Component	Type	Paralellism	Popular Examples
Symmetric Multi-Processor (SMPs)	Scale-up	Coarse	Dell, HP
Massively Parallel Processing (MPP)	Scale-out	Coarse	Teradata, DB2, Oracle
Graphics Processing Units (SIMD)	Vector graphics	Fine	NVIDIA, AMD
Vectorized cores (SIMD)	Vector cores	Fine	Intel AVX instructions

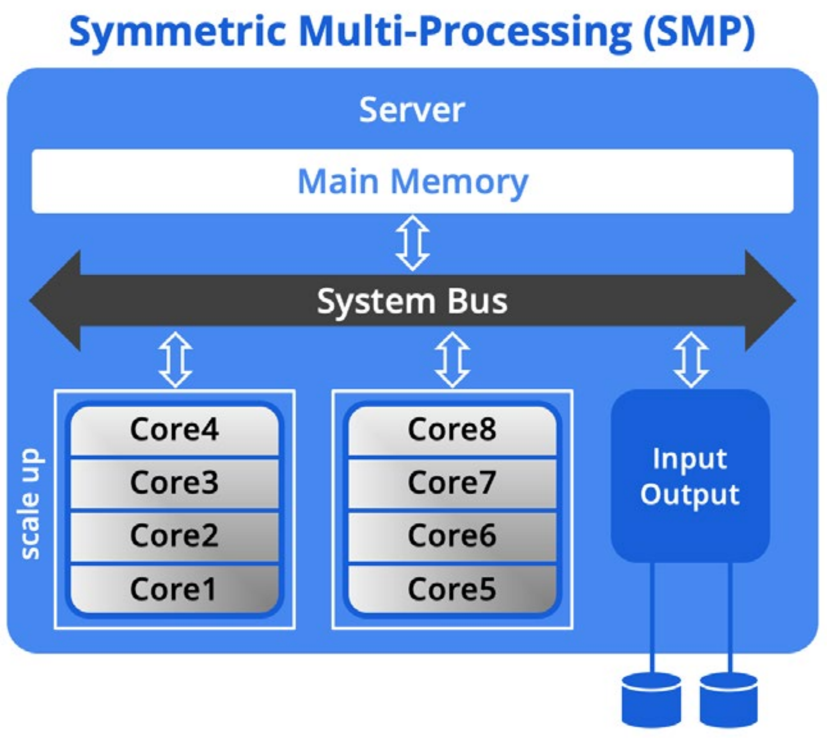
All vendors, including clouds, have standardized on Intel servers as the foundation. That makes software the differentiator. Imagine hardware as ingredients, software as the recipe, and software vendors as chefs. That means some database software is weak, pedestrian fast food. Some are good quality. Exceptional software comes from master chefs. Let's explore how the software utilizes the hardware choices.

Symmetric Multi-Processors (SMPs)

Multi-processing servers are the foundation of all rack servers and cloud instances. In the 1990s, Intel CPUs were single processors installed one per motherboard. Next came adding multiple CPUs to a motherboard. As transistors got smaller, that led to many CPUs on a single microprocessor chip. Each distinct processor inside one microprocessor chip became known as a core. Adding more cores to a server is called scaling-up. Corporate servers could now support hefty workloads with 16-28 cores per server. Each SMP core does completely unrelated tasks such as reading data from disk, sending network messages, or formatting a web page. [Hint: multi-processing is not parallel processing.] Today, for an extra $1000, your PC can have an 18 core CPU. It's not a good investment unless there's heavy-duty software and workloads in mind. Single SMP servers excel at running transactions and collaborative applications. They can do analytics on small amounts of data.

Parallel processing harnesses the performance of multiple cores to solve a single user's task. Imagine a user report asking for 750 million records to be summed taking 200 seconds. Parallel database software can activate eight cores on behalf of one user inquiry. With 8 cores working in tandem, it only takes 25 seconds. Since the cost of cores keeps falling and the performance keeps improving, this is a bargain.

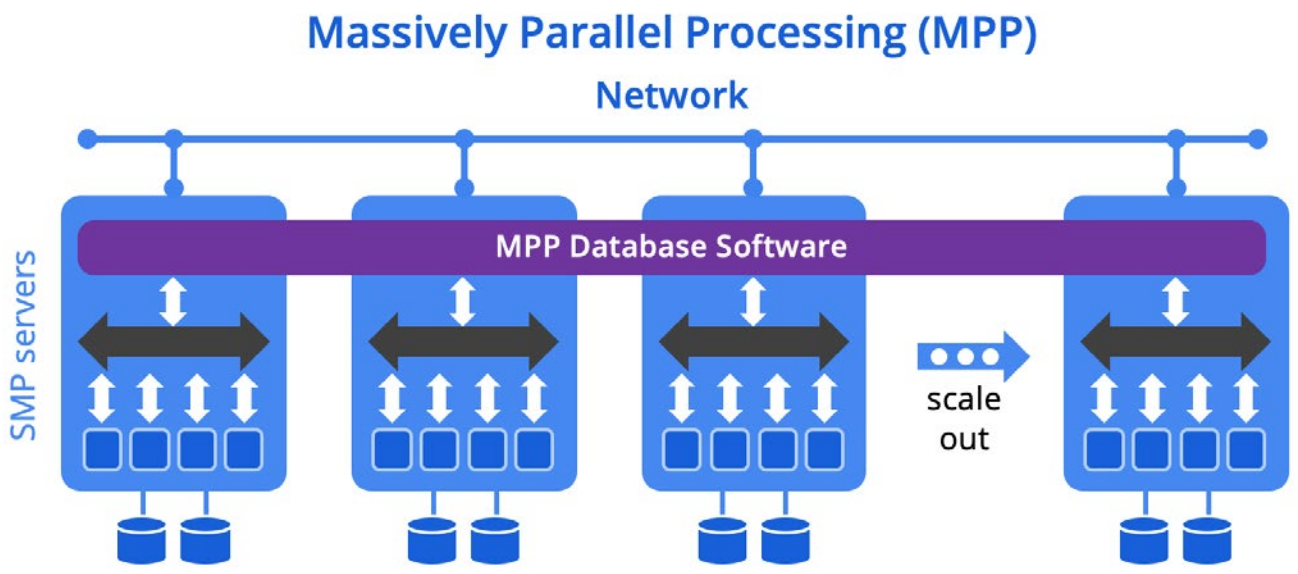
Unfortunately, when SMP cores compete for memory access, some CPUs have to wait their turn. Waiting for a turn millions of times a second means lost performance. This lost performance limits the scale-up capability of SMP servers to between 80-85%. That is, 8 cores yield 6.4X speed-up not 8X. Fortunately, Intel and server manufacturers have been reducing memory congestion for a long time. However, there is a practical server capacity limit when balancing CPUs versus data volume versus workloads. These facts led to the need for scale-out architectures.



Massively Parallel Processing (MPP)

It's possible to attach 100 petabytes of storage to any server. But today's servers can't analyze more than ten or twenty terabytes a day. The solution is an MPP system. These architectures lash together dozens of SMP servers into a single image system. To the business user, the parallel database software appears to be one huge server. Parallel database software adds two unique dimensions. First is scalability limited only by budgets. Second is amazing performance at scale in the 90-98% range. Need more capacity? Add more SMP servers to the MPP database cluster of servers. Today there are MPP databases running 256 servers in a single image. Multiply that by 16 cores each. That's 4,096 cores to grind big data into an answer set. Of course, MPP software is more complicated as is its administration. And MPPs can be expensive. But if the return on investment is higher than 10%, it beats the stock market. Hint: scalable database ROI runs from 40% to 400%.[1]

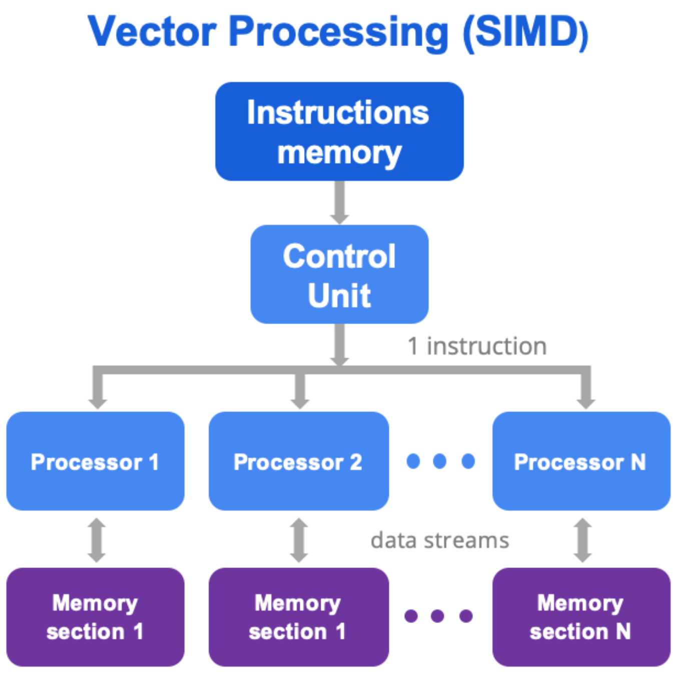
1 Data Warehousing ROI: Justifying and Assessing a Data Warehouse, 2007, https://tdan.com/data-warehousing-roi-justifying-and-assessing-a-data-warehouse/4780

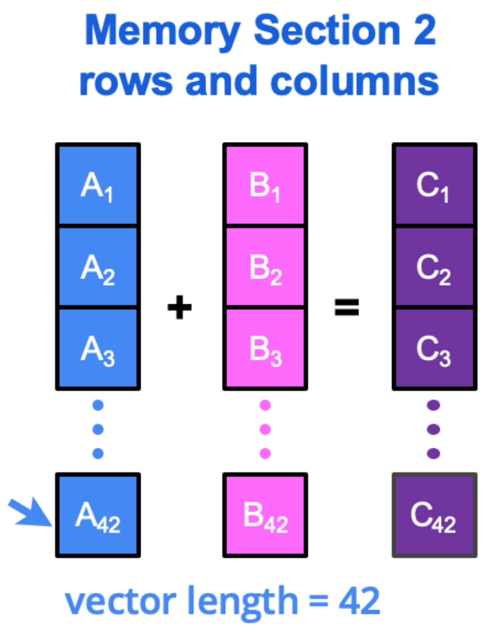


SIMD = Vector Processing

The term Single-Instruction-Multiple-Data (SIMD) is the original name for vector processing coined back in 1966. Cray supercomputers used SIMD vector processing in the 1980s. Back then, vector processing achieved extremely fast performance but required specialized hardware and custom software. The specialized circuits weren't useful or even accessible to commercial workloads. SIMD vector processing simmered in scientific applications for many years. Vector processing is now gaining wide popularity from new software that efficiently exploits and optimizes modern hardware.

As an analogy, a vector is a pointer, like a mouse pointer in a spreadsheet. A vector points to a memory position in a large array of rows and columns. The columns in the memory could be a simple stream of variables (e.g. sensor readings, GPS coordinates). Or the array can be a relational database table (e.g. customer transaction records).




Memory Section 2 rows and columns
A_1 A_2 A_3 A_{42}
+
B_1 B_2 B_3 B_{42}
=
C_1 C_2 C_3 C_{42}
vector length = 42

SIMD = Vector Processing (cont'd)

Vector processing is like an orchestra. The control unit is the conductor, the instructions are a musical score. The processors are the violins and cellos. Each vector has only one control unit plus dozens of small processors. Each small processor receives the same instruction from the control unit. Each processor operates on a different section of memory. Hence, every processor has its own vector pointer. Here's the magic: that one instruction is applied to every element in the array. Only then does the processor move on to the next instruction. This is why SIMD is often called "data parallel" processing. Vector instructions include mathematics, comparisons, data conversions, and bit functions. In this way, vector processing exploits the relational database model of rows and columns. This also means columnar tables fit well into vector processing. But rows and columns must first be organized in memory to align perfectly with processors needs. That means A, B, and C columns must often be side-by-side as in the diagram.

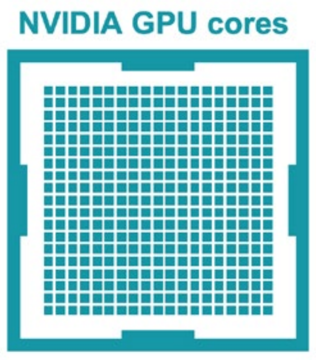
Vector processing doesn't do general purpose tasks. What vectors do best is stepping through a large block of memory with a serial list of instructions. Like the conductor in the orchestra, the entire score is completed before any other music is considered. In contrast, the general-purpose CPU cores are like a Starbucks barista. Every task is different, every moment means running in yet another direction. The data is always different, each decision spawns another decision to make [branches and loops]. The methodical way vector processing steps through memory lines up well with relational set processing. That's why vector processing excels at:

- Machine learning [long iterative chains of math]
- Compression/decompression [images, database blocks]
- Cryptography
- Multimedia: audio, video [CGI]
- Speech and handwriting analysis
- Databases [sorting, calculations, aggregations]

SIMD/vector computers are in two popular technologies today: GPUs and Intel CPUs.

SIMD in Action: Graphics Processing Units

GPUs emerged in the 1990s to enhance Sony PlayStation and Microsoft Xbox graphics. In real time. That's critical for gamers. By the early 2000s, every gamer had a GPU graphics card attached to the motherboard of their deskside PC. The general-purpose Intel CPU now had a buddy on-board: the parallel processing GPU.



GPUs have 100s of small processors etched onto a single microprocessor chip. With this design, NVIDIA soon dominated the PC video card market. It wasn't long before programmers started eyeing the GPU for business purposes. In 2007, NVIDIA responded by creating the CUDA parallel programming language. CUDA provided an abstraction layer above the GPU hardware to make programming easier. NVIDIA now provides dozens of CUDA vector libraries. They include scans, sorting, iterators, math functions, parallel algorithms, and machine learning algorithms.

Today's NVIDIA V100 GPU has 5,120 [CUDA] cores. For certain workloads, GPU parallel vector processing far outruns the performance of general-purpose servers. But GPUs still depend on general purpose Intel cores for numerous database tasks. Thus, there is no sensible way to compare 5,120 CUDA cores to 28 Intel cores. They do completely different workloads.

"Finally, after decades of research, deep learning, the abundance of data, the powerful computation of GPUs came together in a big bang of modern AI."

Jensen Huang, Founder and CEO, NVIDIA
https://bit.ly/33hgb2W

GPUs excel at parallel:	What a GPU cannot do:
Real time data streams	Read-from or write-to disk storage
Neural networks	Send or receive network messages
Graphics rendering	Post business transactions
Natural language processing	Query databases

Use of GPUs was historically limited to pixel-triangles processing: gaming video cards and Hollywood studios. But as programming tools and skills increased, that all changed. GPU devices are approaching commodity status because of their availability in PCs, servers, and public clouds. A lot of corporations don't purchase GPUs. They simply use NVIDIA GPUs in AWS, Azure, or Google cloud servers. This makes prototyping and production use of GPUs accessible for everyone. Use of GPUs skyrocketed in the last five years because of improvements in the programming tools. Today, NVIDIA GPUs power self-driving cars, bitcoin calculations, simulations, and CGI movie production. Many data scientists are programming the GPU machine learning in their deskside PC. Legacy databases don't use GPUs but there are a handful of GPU database startups. If you go to the NVIDIA conference, be sure to stop by the pop-up easels. That's where graduate students explain 100s of new applications enabled by GPU performance.

Intel: SIMD Competition is Good for Customers

"Why Vectorise? Multi-threading and vectorisation are each powerful tools on their own, but only by combining them can performance be maximised. Modern software must leverage both Threading and Vectorisation to get the highest performance possible from the latest generation of processors." Intel https://intel.ly/339Mtgv

For a long time, Intel was also building low-cost GPU graphics to embed on PC motherboards. In 2011, they gave us a surprise. Intel added "Advanced Vector eXtensions" to mainstream processor cores. Those original AVX instructions evolved into the Skylake AVX-512 microprocessor in 2015. AVX-512 means 512-bits are processed for each operation. Intel CPUs began doing vector processing in main memory without a GPU. Intel cores have some advantages over GPUs. First, there was no need to copy data to and from a peripheral GPU device. Second, every rack server and cloud server in the world now would have vector enabled processor cores. Intel had re-established their incumbent vendor position. But what about 28 cores being less than 5000?

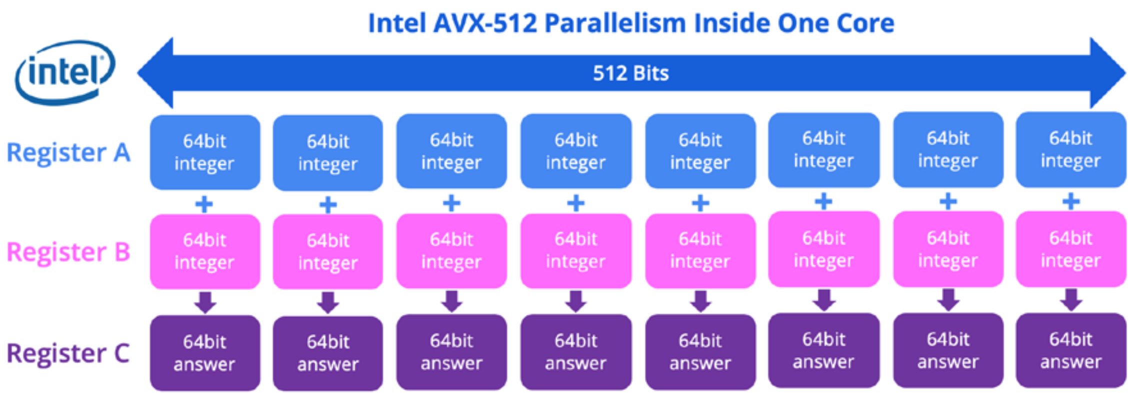
Intel's 512-bits vectors add another layer of parallelism. The 512-bit registers can be logically segmented into 8, 16, 32, or 64-bit slices. Each vector instruction is then applied to all those slices at once.

In our illustration, a single calculation runs eight times faster than executing a single instruction at a time. There can be 28 Intel cores racing through 28 memory segments simultaneously. And each instruction is crunching eight giant integers. That's 224 calculations per CPU clock tick.

On one server in an MPP server farm. Caveat: some cloud vendors offer Intel CPUs or proprietary ARM CPUs. Some ARM cores have vector processing, some do not.

Data management vendors have spotty records for exploiting vector capabilities. Most don't support GPUs at all. Customers have to wrench information from the vendor only to find the use of Intel AVX is sparse at best. A few MPP database vendors exploited the AVX instructions for data compression. Others use AVX for record selection (filtering). Corporate IT organizations have even less success. This is because they can't find a programmer who can use the AVX instructions. Corporations must rely on database vendors for arcane technical skills.

Today's Intel CPUs can run an enormous number of vectorized calculations per second. But feeding the Intel CPU data in the right format comes first. Otherwise, the benefits of parallel processing slow down. That's why columnar data is the starting point. And driving parallel efficiency is still an art form learned by constantly embracing customer demands. And finally, hiding the complexity by making vectors easy to use is necessary for widespread adoption. Here's how it's done.



Kinetica: Four Kinds of Parallel Performance

Kinetica's columnar database was born to exploit three kinds of parallelism: SMP multi-core, MPP scale-out, and GPUs. Intel AVX was added later when the first AVX processors became available. That's more parallelism than any other database technology, often three more. Some legacy databases use a limited set of Intel AVX instructions for data compression or record filtering. In contrast, Kinetica fully exploits both AVX and GPU vector processing in aggregations, windowing functions, filters, joins, sorting, GIS functions, rendering, graph analysis, and machine learning. This is Kinetica's forte, their raison d'etre: exploiting vectorization in every possible function.

So how fast is vector processing? Intel claims 16X speedup in their blog.[1] Kinetica says:

- A financial services firm ran a query on a 700-node Apache Spark cluster. It took hours on Spark but only seconds on 16 Kinetica servers.
- A world-renowned retailer reaped large cost savings. They consolidated 100 Cassandra and Spark servers down to eight Kinetica servers.
- A pharmaceutical company got identical performance between an 88-node Hadoop cluster and a 6-node Kinetica cluster in Azure.
- One large telco company realized their current system would take six years to render all their maps and location data in order to optimize their network for 5G rollout. The Kinetica platform did it in 50 minutes.[2]
- A retailer ingests 250 million records per day for inventory tracking on an 8-node Tesla P100 Kinetica cluster while performing 112,000 read queries per second.[3]
- A large mail carrier ingests a billion events per day for mail routing optimization on a 10-node P100 Kinetica cluster, while 15,000 users are simultaneously querying the data.

We know that performance varies by workload. Still, it's clear that vectorization has undeniable performance benefits for analytics workloads.

1 Intel, Vectorization: A Key Tool To Improve Performance On Modern CPUs, 01/25/2018 https://software.intel.com/content/www/us/en/develop/articles/vectorization-a-key-tool-to-improve-performance-on-modern-cpus.html

2 Rebecca Golden, Helping Enterprises Say Goodbye to Static Analytics and Hello to the Power of Active Analytics.

3 Kinetica Dipti Joshi, Real-Time Streaming Analytics with Kinetica, December 4, 2019; https://www.kinetica.com/blog/real-time-streaming-analytics-with-kinetica/

Reducing Complexity

Kinetica's first customers back in 2010 pushed them into grand challenge vector processing. Tough customers with near impossible workloads kept Kinetica software engineers laser focused. With more than a decade of grand challenge clients, it would be wrong to call Kinetica a start-up. Kinetica outgrew that years ago. And their customers came back for more. Hold your applause, it gets better.

Kinetica's vision includes making all parallelism transparent, deep inside the database. Exploiting GPUs or Intel AVX requires no effort by users or programmers. Dozens of database functions automatically exploit available vector processing devices. There is no need for complex programming. Line of business analysts with BI tools invoke GPUs or AVX vectorization without knowing it. Kinetica software fetches the data, organizes it for vector processing, and invokes the local programmer's UDFs. Competing GPU databases leave all that to your programmers. Kinetica is a departure from startup quality "GPU databases" which are often difficult to program. For most tasks, there is no programming required.

SQL is still the lingua franca of analysts, programmers, and popular software tools. Imagine a single cloud instance with four upscale NVIDIA GPUs attached. Kinetica converts SQL to vectorized operations for you. Kinetica runs optimized SQL data retrievals and marshals the data into each of the four GPUs. No programming required. Putting all these parallel services together yourself is complex work. Perhaps your IT department put all the tables into Parquet cloud objects. No problem. Kinetica transforms cloud objects into GPU or AVX friendly formats for you. Most GPU databases don't do much or any of this. Furthermore, Apache Spark CUDA coding is –to be kind – also challenging.[1] If needed, programmers can write User Defined Functions (UDFs) that run CUDA or AVX (e.g. Tensorflow) within Kinetica database. UDFs are well understood by database programmers. And they add a lot of infrastructure for CUDA and AVX to reduce the programming effort for developers.

1 Deep Dive into GPU Support in Apache Spark 3.x, Jun. 27, 2020 https://databricks.com/session_na20/deep-dive-into-gpu-support-in-apache-spark-3-x

Kinetica Analytics

Some in-database functions are ideal for vectorizing. Graph analysis, time-series, and geospatial[1] are examples. Imagine a complex graphic chart of 10 billion geospatial records to display in ESRI, Tableau, or Qlik. The MPP cluster then ships 10 billion records to a business intelligence (BI) tool overwhelming the network. Most BI servers and PCs lock-up for hours. Or worse. Kinetica uses SQL and vector processing to analyze the 10 billion rows fast. Calculations and visuals come back in seconds, not hours. The geospatial result is rendered server-side in a small 5-20MB graphic image. Kinetica ships the image to ESRI or Tableau using vector-tile-service (VTS), web-mapping-service (WMS), or Geo-JSON format. If the analyst drills down into the visualization, Kinetica recalculates the results, sending back a small visual map again.

Another built-in Kinetica capability is graph analytics.[2] Graph as in vertices and edges. It's surprising how well graph analysis fits into vector processing iterations. There are a few graph databases like Neo4J (SMP only) and open source like TigerGraph (MPP). Neither use SQL meaning they require specialized skills. They have little support from business intelligence and data integration tools. In contrast, Kinetica built graph analysis into their database kernel. It's not a bolt-on side-car or an orphan graph database. Kinetica made graph analysis, geospatial, and machine learning a native part of their database.

1 Kinetica Geospatial Visualization for Tableau, https://www.kinetica.com/events/tableau-geospatial-extension/

2 Kinetica Graph Analytics: Multiple Supply Demand Chain Optimization (MSDO) Graph Solver, February 9, 2021, https://www.kinetica.com/blog/kinetica-graph-analytics-multiple-supply-demand-chain-optimization-msdo-graph-solver/

Last, be aware that Kinetica is partnering with Confluent's Kafka to support customers with heavy streaming analytics workloads. Confluent has many streaming analytics partners and tools available. Kinetica, however, adds a couple benefits that are hard to pass up. First, there are many situations where the data must be captured in storage at real time speeds. Surprisingly, Kinetica's data ingest speed matches the incoming streams and outruns the best legacy databases. This capability was driven by Kinetica's defense customers, such as the NORAD. [Talk to Kinetica's CTO and chief architects to find out how it's done.] Second, there are many situations where streaming data is not enough data to provide decision support.

Imagine dozens of manufacturing IoT sensor streams blasting into the data center. The real time analysis shows two high profile manufacturing machines will break down within the hour. What the IoT streams cannot tell us is whether the repair parts are in local inventory. The IoT stream also can't help us determine if the right repair person will be available within an hour. This is where coupling a database with a high-speed streaming service like Confluent can provide the full context for decisions. Kafka/Confluent is a great tool for streaming analytics. But sometimes a database is required for a 360º view of the decision. Kinetica's architecture provides real-time ingestion with all the benefits of a database: data fusion, historical data, and analytic strengths. Furthermore, where else can you find Kafka streams of time-series data feeding geospatial and graph analysis?

Conclusions

Kinetica's forte is its relational foundation supporting vector processing in both SMP and MPP server configurations. It's ideal for high-capacity data, real-time streams, or intense number crunching workloads. Data scientists will be euphoric with a small dedicated Kinetica "data lab." It can do web page click analysis, IoT sensor stream analysis, and supply chain analysis. But Kinetica can also take organizations into the digital transformation future. That's where the bold are combining geospatial, time-series, and graph analysis to solve previously impossible business problems.

About Kinetica

kinetica

Headquarters

101 California St, Suite 4560
San Francisco, CA 94111

(888) 504-7832
+1 415 604-3444

To learn more

visit us at
kinetica.com

or
contact us