All Your Data at Lightning Speed: In-Chip® Technology Revolutionized Analytics

Introduction

Cast your mind back a moment to your last holiday. Remember the weather? The food? The dreaded packing decisions?

Perhaps you're someone who prefers to pack super-light, in a small bag that doesn't slow you down - even though that means sacrificing on stuff that you might actually need when you get there?

Or are you one of those people that can't bear to leave anything behind, and ends up dragging every conceivable item you might need with you in a series of ginormous suitcases - even though you waste endless time just rooting around for whatever you're after, and only end up using a fraction of that stuff in the end?

In the past, this was the core conundrum BI users faced when choosing between using RAM or disk-based storage to load and process queries.

Either you could select a system that relied on your machine's RAM for data storage, giving you speedy access and processing, but limiting the amount of data you could have readily available at any one time...

...or you could opt for disk storage, which typically means you can access far more data and files but have to wait for an absolute age for the system to painstakingly sift through all the stuff you don't need, in order to find what you do.

Not exactly a great choice, huh?

Especially since the latter didn't just mean more wasted time, but also a far greater financial outlay for hardware that could actually process those big, data-intense queries.

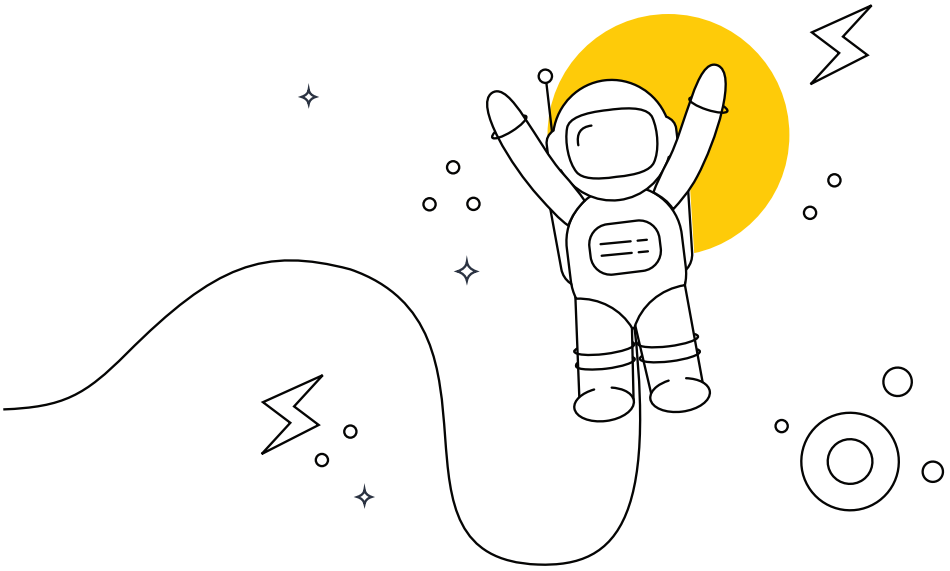
But then along came In-Chip® processing, which combined the best of both approaches, while actually reducing your processing power requirements - and therefore your costs - at the same time.

Now you don't have to choose between lugging around a huge receptacle of data you probably won't need, or a fraction of the whole that you can dip into easily, but might not be enough. This new system essentially means you get to keep the small, agile option, without sacrificing access to all the data you own.

Imagine it like this: your little hand luggage case is transformed into a Mary Poppins bag. Despite letting you zip around at speed, whenever you put your hand in, you can pull out exactly what you need, as if by magic. And when you don't need it, it isn't there to weigh you down.

Sound too good to be true? Well, that's the beauty of super-smart, modern BI technology for you.

In this whitepaper, we'll dive right in and explain exactly how it all works - and why that's great for your BI needs.



Part One: The Basics

The Problem

As we've seen, business users of BI platforms don't want to wait around for results, nor will they tolerate incomplete or inaccurate reports.

Ideally, you want to be able to query huge data sets across multiple, specialized databases, even running concurrent queries, and still get the answers you need as quickly and efficiently as you would from, say, running a Google search query.

All of this puts a huge strain on commodity hardware, not least because datasets and user numbers are growing exponentially, and requirements change all the time.

Oftentimes, when companies need to handle mammoth dataset challenges like these, they do things like setting up parallel operations, spread across multiple machines - called database clusters.

This isn't great for agile BI, though; it puts way too great a burden on your IT resources and leads to all kinds of bottlenecks and hold-ups.

That's why, when it comes to BI, the emphasis is on creating more efficient ways of deploying simple architecture on a single commodity server, rather than resorting to clusters.

How Computer Data Storage Works

Let's back up for just a moment and look at the two main forms of storage computers have: disk (or hard disk) and random-access memory (RAM).

Modern computers typically have anywhere from 15-100 times more available disk storage than they do RAM. Plus, when you power down your computer, anything stored on RAM is instantly lost, whereas anything stored on your disk is unaffected.

What's more, RAM is vastly more expensive. 1 GB of RAM costs around 320 times that of 1 GB of disk space.

So why not focus on using all that cheap, readily available disk space for your data storage needs?

Well, the trouble is that reading data from disk is much, much slower than reading it from RAM. That's why your computer processes rely on RAM as much as they can - it would be agonizingly slow to do anything otherwise.

Disk-Based vs. In-Memory

Okay, so we've summed up computer data storage in a nutshell...but how does this translate into databases?

Put simply, if your database relies on disk storage, it's a **disk-based database**. If it relies on RAM, it's an **in-memory database**.

So: disk-based databases query data that resides on the hard drive. They work on the basis that the total amount of data available can't possibly be loaded into RAM, as there's just not enough space there.

Like our giant suitcase analogy, they let you keep hold of every tiny item you might possibly need...but without an expensive, high-powered way to sort through all that stuff, you're looking at seriously slow disk operations. This just doesn't cut it for most modern BI needs.

In-memory databases, on the other hand, load the dataset into memory (RAM) before querying it. That means you can whizz through the data at lightning speed, but there's an obvious drawback: if you're dealing with a big dataset, you need a LOT of (hugely expensive) RAM. Without that, you'll need to limit your speedy in-memory database to handling much smaller datasets; your "carry-on bag" of data, which could be woefully incomplete.

Part Two: The Evolution of BI

As we've seen, taken on their own, neither in-memory nor disk-based technology can offer a perfect solution for BI users.

In fact, we can see this brought to life through the ways in which BI solutions have evolved. Let's take a look at this in detail.

First-Generation BI: Disk-Based Databases

Developed in the 1980s, the first generation of BI technology had to work with the hardware that existed at the time, which had very little RAM, relatively weak computer processing units (CPUs), and limited disk space.

Back then, BI solutions were based on relational database systems and geared towards straightforward, transactional processing: inserting, updating, and deleting records that were stored in rows. If that's all you want from it, it still does just fine.

Not only do these require slow disk reads (for all that data stored on disk), the way the tables are designed means the process eats up RAM, making it hard to use this efficiently. The system's single architectural approach just isn't built to accommodate complex requirements like aggregating, grouping, and joining.

And then there are issues with the Structured Query Language (SQL), that's used to extract transactions from relational databases. This is designed to be great at fetching rows, but most of the time you don't need the whole row of data for a BI query! Rather, you probably want to simultaneously retrieve lots of partial rows and apply complex calculations...but it's practically impossible to formulate an efficient BI query like that using the syntax of SQL.

When you try to apply this technology to high-performance BI on large data sets, though, you have a problem.

Second-Generation BI: In-Memory

Clearly, a new approach was needed. This leads us to the next development in BI technology: in-memory databases.

As we've seen, these tackle the problem by shifting the whole dataset into RAM, meaning you don't have to dip into the disk to query the data. That immediately and immeasurably boosts performance, but it does mean you need a ton of available RAM (or not very much data).

Admittedly, there are some things you can do, like employing compression techniques to squeeze more data into the RAM you have, but that doesn't really fix the problem. Limited memory space inevitably reduces the amount of historical data you can include, and the number of fields you can query, impacting on the quality and effectiveness of your BI.

Adding more and more RAM is possible, but it means paying exponentially more for your hardware. Typically, your run-of-the-mill, desktop-class computer with standard hardware physically supports up to 12 GB of RAM. If you want to move to a system allowing up to 64 GB of RAM, that will double the price tag. Moving beyond 64 GB requires a full-blown server, which is very expensive.

If you have a lot of people vying to use the BI application at once, you're in even more trouble. Just 5-10 people using the same in-memory BI system will quickly double the amount of RAM you need to generate query results; if you want to scale up and roll this out to many users across the organization, it's cash or crash, unfortunately.

And then, of course, you have the problem of reloading the whole dataset into memory every time you reboot your computer, which can become so time-consuming it's hardly better than sticking to disk.

So, in-memory alone isn't enough to fix BI's data loading problem. Rather, you need a smart solution that combines the best of disk and RAM...and that brings us to In-Chip® technology.

Part Three: In-Chip® Technology

How Does In-Chip Technology Work?

Invented in 2010, In-Chip technology, made up of several pending patents, draws on everything that's great about in-memory while tackling its cost and scalability issues.

It does this by efficiently making use of the best qualities of modern hard disks, RAM and CPU, in real time, allowing you to access the maximum possible storage capacity (from disk) while ensuring performance that's equal to or even faster than in-memory performance. Plus, you can run many concurrent queries without overloading the system.

By optimizing technology that already exists on standard commodity hardware, particularly the CPU, you get this vastly improved performance without having to splash out on more computers or processing power, too.

Did You Know...?

In-Chip® is a registered trademark of Sisense, and the solution is made up of several technologies that all have pending patents.

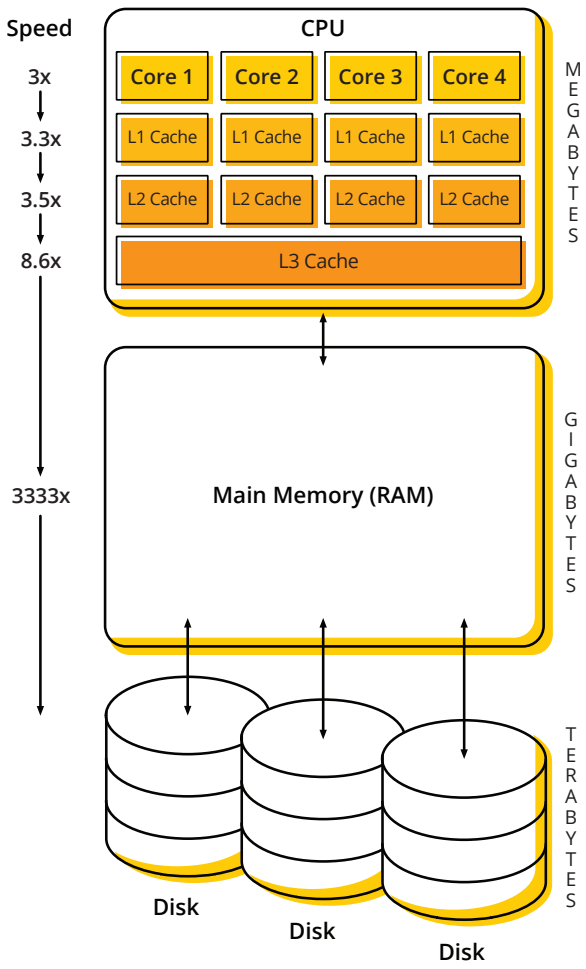
CPU Utilization

Put simply, In-Chip uses its own code to optimize how RAM is used and how the technology communicates with the CPU, instead of relying on the operating system to do that.

Importantly, it dramatically cuts down the number of times the same piece of data gets copied between RAM and the CPU, by referring back to the cache wherever possible. It does this by optimizing kernel-level instructions for analytical BI questions and then optimizing the intermediate results for CPU cache sizes, across all cache levels. The resulting query performance is ten times better than traditional in-memory systems.

By loading compressed result sets into the cache, and having decompression operations read and write to cache, the application runs incredibly fast, without creating a huge memory footprint.

This makes the whole system more efficient, allowing performance to soar without demanding extra RAM.



Storage Utilization

Here's where it gets really interesting.

In-Chip stores all its sprawling datasets on disk, but instead of then loading the whole thing into RAM, it only scoops up the parts of the data you need to query in real time. You can imagine what a difference that makes to the amount of RAM you need.

Once you're done, it either releases this data back onto the disk - or, if usage stats suggest that it makes sense to keep it local, this will stay in the RAM to serve other users running queries, too.

All of this is made possible using a columnar database, which stores information in columns rather than rows of data. This means that In-Chip can speedily scan fields on the disk without having to go through the whole table.

For example, if you have a table made up of 10 fields and 10,000 rows, a tabular database would need 100,000 disk reads to scan this. A columnar database can scan this in 10,000; i.e., a tenth of the time.

Another perk is that columnar databases allow you to compress data easily, which saves storage space - and, more importantly, In-Chip can perform calculations on these datasets without having to decompress them first. This reduces the RAM requirements even more.

Concurrency Handling

Another savvy feature of In-Chip is that it speaks the language of the columnar database, rather than relying on a SQL-based query engine that, as we've seen, isn't optimized for picking just the bits you need out of a huge database.

Typically, when you run an SQL query, only the result of whole queries are saved, so this is only useful if you run exactly the same query again.

When you query a database using columnar algebra, on the other hand, this query is broken down into thousands of small instructions. If someone else makes a different query at the same time that contains some of the same instructions, it can reuse them, increasing the performance of all these concurrent queries at once, instead of slowing them down!

Picture it like this: you're in a busy restaurant with only one chef. Each time a table makes an order, a waiter takes the ticket to the kitchen, and it gets added to the queue. The chef doesn't look at the next ticket until she's finished the order beforehand, even though she might be making 10 variations in a row of the same order.

Now imagine that, instead of a series of tickets, the waiters are able to combine all their simultaneous orders and simply tell the chef that, collectively, they need 10 risottos, 7 pastas, and 5 salads. By bundling these requests together instead of starting again each time, the chef works much faster, and once they're ready, the waiters simply recombine their orders and deliver them to their table. In that way, the chef can actually serve 10 tables just as fast as she could usually serve, say, 2 or 3.

In essence, that's the difference between handling SQL queries, where every new query adds to the pile, and In-Chip columnar algebra, where users can make concurrent queries without demanding much more RAM, or noticeably slowing performance.

Final Thoughts: The Business Benefits

Why Do I Need This?

That's all very clever, you may be thinking, but what's in it for me?

In short, it means you never have to sacrifice between in-depth, granular insights and performance. There's less data prep involved and less reliance on IT, meaning non-techie staff can get rapid, business-critical insights in real time, without those frustrating bottlenecks.

Right now, most BI solutions can handle a maximum of a few tens of millions of rows without IT having to get involved. But as you no doubt know, your datasets are ballooning all the time. Even if that works for you now, in the near future, this might not be enough.

What's more, you get better quality, faster analytics without having to splash out on hardware.

In-Chip technology, on the other hand, can handle billions of rows while barely batting an eyelid. What's more, it automatically adjusts to fit the data volumes, user numbers, and workloads that are running.

You don't need any special hardware - and if you do run out of resources, you can just add another commodity server and split your workload between them. It's not going to cost the earth.

Who Uses It?

Don't just take it from us, take it from our clients.

Here are just two examples of happy customers from the thousands using Sisense's In-Chip technology.

First, Magellan Vacations is a luxury hotel booking company that relies on telephone-based agents to provide its clients with personalized recommendations and book hotel rooms.

To support its agents and track performance, the company needs to track huge volumes of sales metrics, such as closing rates, commissions, and bookings by destination.

The company tested an in-memory technology, but the performance was sub-par, and the solution required IT specialists to work with the tool's proprietary scripts, as well as IT consultants to work with and modify the application.

Instead, Magellan Vacations selected an In-Chip technology solution that gave agents near-real-time feedback on sales closings, destination performance, and other metrics that would help them better serve customers. Because the solution was so easily scalable, the company didn't need a major infrastructure upgrade or investment in expensive IT resources.

Or look at Wix, the enormously popular website builder.

Supporting 22 million websites, the company needed powerful analytics and reporting solution that could help them swiftly track metrics like conversions, marketing campaign efficacy, and user behavior.

Before they switched to In-Chip technology, data was largely managed via scripting, and reports were difficult to explore and change. Now the team can swiftly combine sources, analyze behavioral data and generate reports that validate the success of marketing campaigns and changes in product user behavior.

Upgrade Your Analytics

Choosing In-Chip technology means opting for unparalleled scale and performance, making it possible for companies to roll out a genuinely self-service data environment without the associated costs.

By providing speeds of between 10-100 times that of in-memory solutions while offering unfettered access to all data stored on disk, you can continue to glean crucial insights that take in the whole of your rapidly growing datasets, in granular detail, and consolidate reports that drive business decisions and performance, all across your organization.

Want to check out Sisense's In-Chip® technology for yourself?

CLICK HERE TO TEST DRIVE SISENSE ON YOUR OWN DATA