www.inzata.com

Inzata

Core Aggregation Engine Overview

Version 1

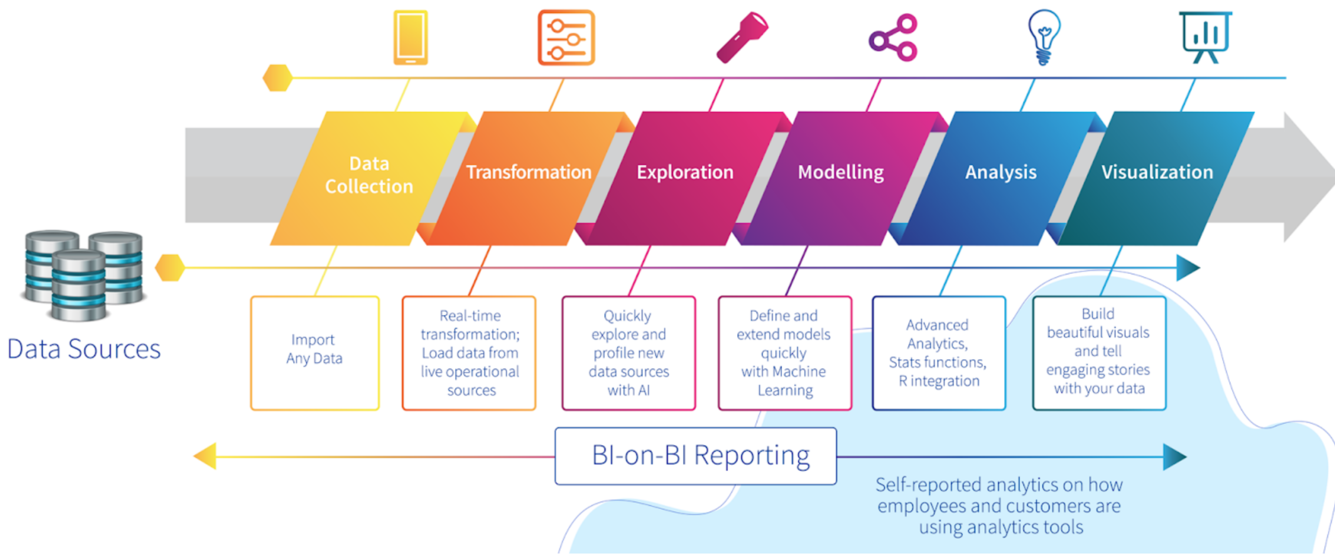
December 2021

CONTENTS

Inzata, LLC
5401 W Kennedy Blvd
Suite 100, Tampa, FL

Inzata Platform

Inzata provides a powerful, efficient yet affordable Big Data Analytics platform in the cloud for ad hoc report generation and processing on complex, enterprise-wide data sets. Inzata's unique back-end architecture and middleware allow integrators to provide significantly faster, highly scalable, multi-user, multidimensional data analytics using commodity hardware. The API and generated AQL high-level language allow the creation of ad-hoc self-service reports.



Inzata provides you with real-time analysis of Big Data on the cloud because it:

- Is built for real-time ad-hoc reporting on structured data
- Uses distributed computing (MapReduce, distributed storage, cloud platform, etc., similar to Hadoop) with massive parallel processing on Big Data in cloud configuration
- Benefits from integrated BI Middleware with Inzata query execution engine for optimal performance
- Maintains linear scalability for a large number of users and projects with complex data models
- Is built for fast BI project deployment with automated modeling data integration and enrichment functions
- Has its UI modules while also being able to integrate with 3rd party apps via REST API library functions and Inzata Widget Web objects

The Inzata Engine has built-in its advanced integrated business layer:

- BL is optimized for ad-hoc analytics and near real-time reporting
- It is designed for OLAP supporting unlimited dimensionality of business logical data models for:

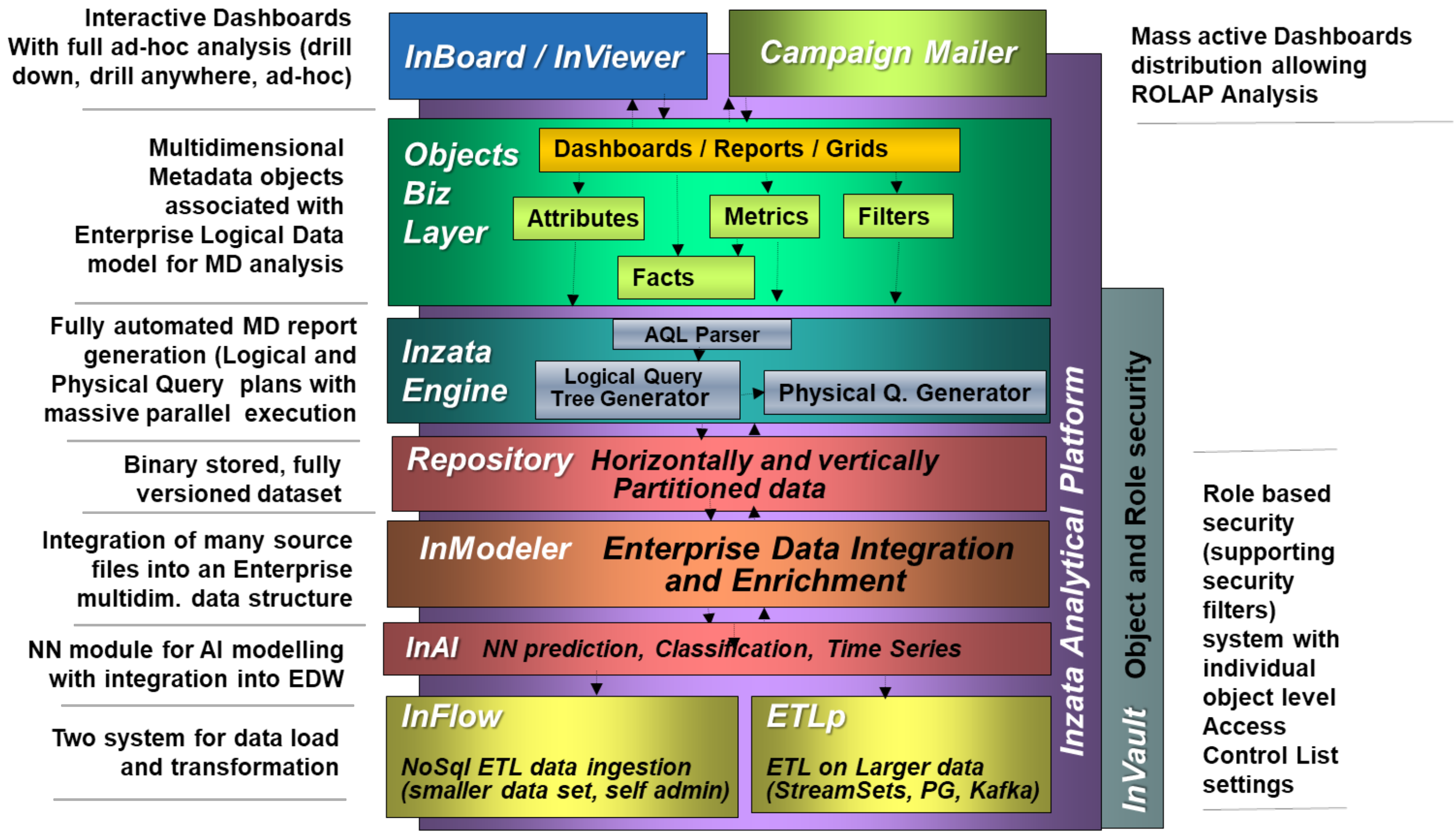
- Reporting and drilling
- Metric complexity – embedded aggregation with dimensionality using recursive metric definition allowing to design general queries w/o limitation
- High level (“SQL Like”) Analytical Query Language optimized for dimensional analysis
- Fully automated generation of complex queries without additional layers typical for current ROLAP systems
- Additional report retrieval speed achieved by multidimensional intelligent caching (extensive caching of both metric results and constrain masks)

Inzata Platform Architecture

The Inzata BI platform provides a comprehensive solution to current challenges for BI tools and environments:

1. Speed and flexibility of Ad-Hoc reporting
2. “Big Data” analysis
3. Scalability in terms of number of concurrent users
4. Complexity of business data model in a current dynamically changing environment.

The platform provides a comprehensive solution that consists of fully integrated components representing the entire BI stack. The diagram below describes the core Inzata architectural modules and components:



Inzata Engine

BI Analysis for large amounts of data and the rapidly increasing number of users are representing challenges for business intelligence vendors as well as database engine vendors. Traditional HW concepts (Row-based databases, multidimensional OLAP approaches utilizing current HW) do not provide sufficient speed and scalability to respond to current BI challenges. Our BI system is based on almost 10 years of development experience with BI tools. The heart of the new, fully innovative, In-Memory Aggregation system is the HW accelerated aggregation engine used in-memory aggregation. This concept represents a unique solution based on maximal utilization of CPU HW resources. The CPU accelerates BI application performance by massive parallel execution of the compute-intensive tasks of the in-memory data aggregation process. There are 2 significant benefits of this unique, state-of-the-art design:

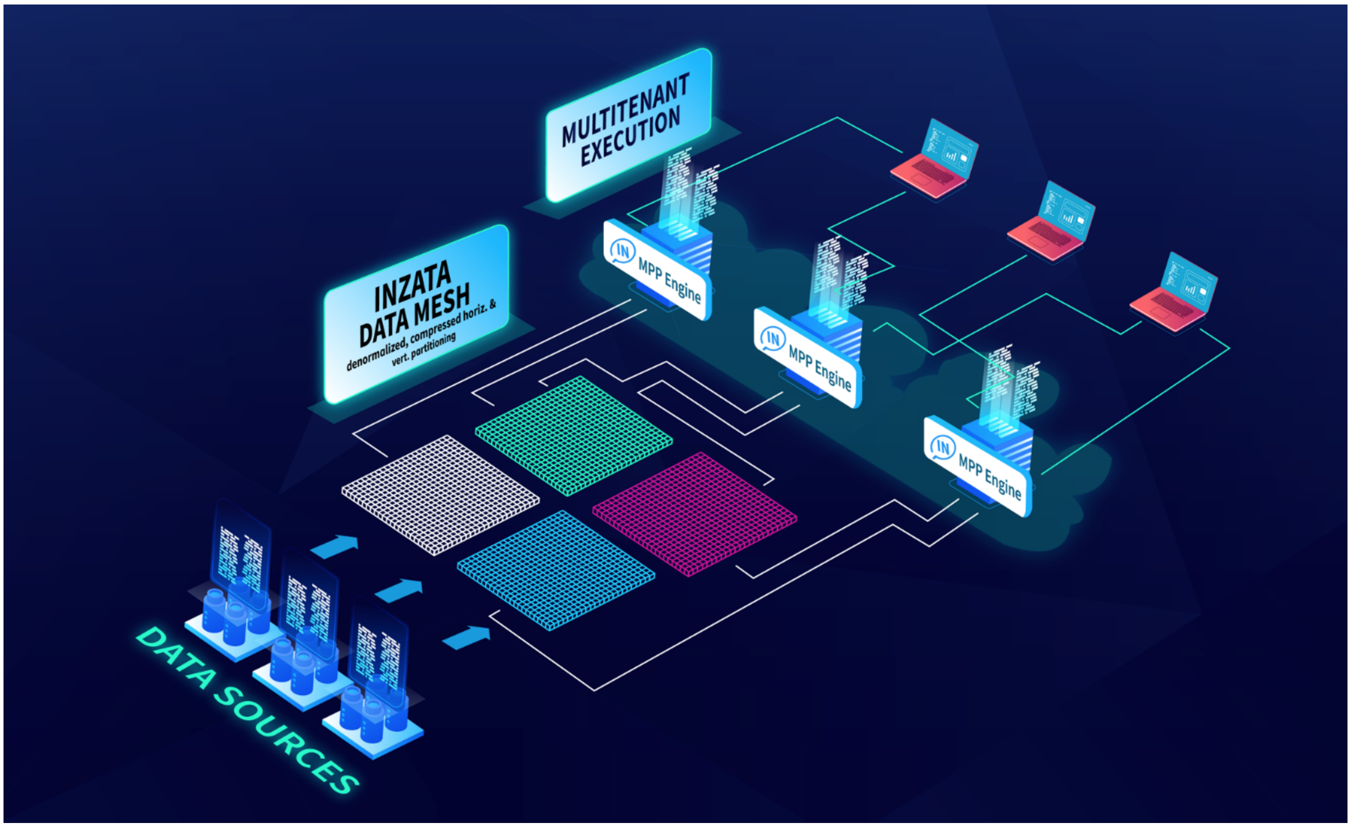
1. From a user's perspective, the application runs significantly faster because it's using the massively parallel processing power of the CPU to boost performance of business intelligence analytics and reporting over a large amount of data. The bottom line of this solution is the ability to achieve extraordinary performance on commodity hardware.
2. There is no need to compromise analytics depths and breadth by minimizing data size like in a traditional OLAP system. This new design allows you to execute drill-downs and drill anywhere on virtually any data breadth and depth (1 mil. rows+)
3. Massive parallel processing of a single report as well as reports coming from multiple users.
4. Inzata Features and Benefits – execution engine and business layer:

Key feature	Benefit
○ **distributed**, REST based architecture ○ multitenant components ○ MapReduce, MPP processing ○ utilization of virtual HW in the Cloud ○ multidimensional caching	➔ $$ Saving on overhead cost ➔ superior Price/Performance of the true cloud solution ➔ ability to run many projects on one HW cloud system ➔ scalability in number of users and data volume ➔ significantly faster than standard report execution engines on the market ➔ Big Data Analysis
○ AQL – high level analytical query language ○ multidimensional ROLAP-like analysis ○ reusability and share ability of analytical objects (LDM, metrics etc.) ○ automated data model design from raw data	➔ ability to process very complex reports on complex data models (unlimited number of dimensions) ➔ fast project implementation ➔ low cost of projects administration and maintenance
○ data versioning ○ zero data redundancy ○ partitioned (vert. and horiz.) data repository ○ direct processing of denormalized data ○ no need for indexes or pre-aggregation	➔ query performance ➔ near real-time processing ➔ continuous report delivery ➔ efficient incremental loads ➔ ability to easy modification of logical data model

Inzata, LLC
5401 W Kennedy Blvd
Suite 100, Tampa, FL

Key Feature	Benefit
○ De-normalized data structures ○ REST API ○ Web widgets	→ Automated Data modelling and data integration → flexible integration with 3rd party portals and apps → very fast development of new BI apps
○ core algorithms written in OpenCL	→ ability to utilize virtual cloud HW → prepared for future HW architectures (CPU/GPU/Coprocessors)
○ automated report code generation ○ drill anywhere ○ easy Ad Hoc reporting	→ efficient BI analytics and reporting → fast user training
○ integrated predictive modeling for advanced analytics and statistical calculations	→ embedded advanced statistical models into BI reporting

The Inzata Engine provides enhanced scalability and performance gains that give you unlimited access to deep data. It also provides you data breadth: there is no limit on the number of dimensions, fact tables, or aggregate tables you can build. You are not constrained in your data structure based on the number of source files or in your business model by our product. Essentially, the Inzata Platform with its Engine provides fast access to the data breadth, depth, and reporting range you need to answer your most valuable business questions and help solve your most pressing business problems.



Inzata, LLC
5401 W Kennedy Blvd
Suite 100, Tampa, FL

Summary of Inzata Engine Unique Features

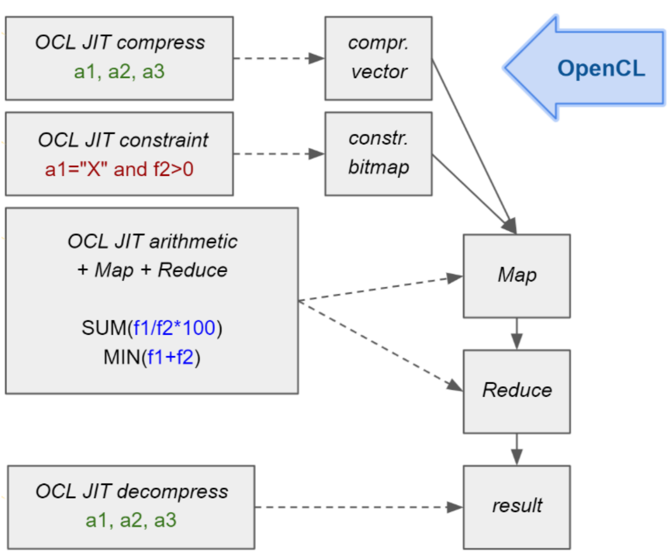
Inzata's BI systems and methods are based upon years of development and experience with BI tools. The first unique feature is represented by a combination of several features that are designed to address/answer challenges of current BI reporting needs:

A) Big data is stored in the file system of a data-crunching computer (not as in a traditional Relational Database – as is common in BI solutions). Further data is stored in a columnar structure and it is fully horizontally and vertically partitioned to allow it to be processed in a fully distributed environment (computer clouds).

B) Due to optimized data storage (not in RDBMS), a new special Analytical Query Language ("AQL") has been designed and developed. It has a clean design without any necessary exclusions dictated by the architecture of data storage (i.e. like other solutions - materialized multidimensional cubes), it is optimized for general multidimensional data analysis.

C) Inzata's system has a fully integrated and optimized business layer with query execution and the backend system features: a closely integrated AQL language and an Inzata Query Tree decomposition engine (not a general-purpose query and aggregation engine).

D) An automated query generator provides Logical Query decomposition (from the Logical Data model). The user's report query is automatically decomposed to a tree of the logical multidimensional objects, each defining a specific subspace of the whole multidimensional space. This query can be optimized by pruning using algebraic and heuristic rules. From a logical data model point of view, the query decomposition represents the report's image in the overall multidimensional space. This highly optimized logical query decomposition system is tailored for denormalized data stored in columnar storage.



E) An automated query generator that provides transformation of multidimensional subspaces into Macro Language – representing the physical execution plan of the projection of metrics into multidimensional subspaces defined by an ad-hoc report. Due to the fact that macro definitions are fully controllable, the system can easily be enhanced to deploy other special processing tasks (e.g. statistical tasks, post-processing tasks for customized reports' display, etc.). The system is not dependent on predefined sort keys materialized in advance. The Macro code is transformed into the hardware-accelerated execution code (utilizing the Just-In-Time compilation approach with the OpenCL compiler). Inzata's solution takes advantage of massive parallel processing within a single node of the Inzata cloud. This is also a unique system, due to the fact, that there is no other automated query tree generator, that would be automatically generating queries optimized only for data stored in columnar way residing in the file system.

F) Fully integrated optimized business layer with query execution and backend system having features such as a multilevel caching system which allows re-use of previously calculated reports and their temporary results. Additional report retrieval speed is achieved by extensive caching of metric results, compressed keys, and constrain masks. The system recognizes a given run query (or parts of a query) that have already been calculated. Instead of re-executing the entire query the results or a part of the results are pulled from the cache. (No other multilevel caching system on the BI tools market can automatically generate and then reuse multidimensional subspaces stored in a columnar way directly residing in the file system of execution work stations). The designed multidimensional data repository is flexible enough to create a huge number of derived subspaces, those subspaces can be utilized not only by whole reports but also by low-level componential aggregation tasks (SC caches) that are used for data retrieval of completely different reports with different metrics, than reports for which these low-level caches were originally created.

G) In memory aggregation logic is designed to take full advantage of massive parallel HW components and therefore a special implementation of MapReduce concepts has been designed for aggregation subsystem using massive parallel processing.

H) A fully modular processing system is designed for scalable cloud computation with separate dedicated commodity computers for each of the below functions (there has not been any similar cloud environment of in-memory aggregation servers with dedicated, role-based systems running in parallel data aggregation tasks not only in commodity MPP system of a single system but also simultaneously on multiple servers in the cloud. This parallelism allows you to run one single, low-level aggregation task across all the available workers on the aggregation node. To be able to do that, individual nodes of the multi-node K8S environment of the cloud are assigned into their individual roles:

a) **Data Crunching** – Example: Multiple processing workers are each capable of processing data aggregation tasks on all the available cores of the node. There can be multiple aggregation nodes in the given cloud to support scalability requirements for supporting larger concurrent user communities while still maintaining the excellent response time for ad-hoc reporting.

b) **Metadata Serving** – Example of configuration: 3 separate metadata nodes provide extra performance support for the metadata DB to search for business metadata objects. With such separation, there is no performance degradation for all the tasks requiring access to the Inzata MD objects. These 3 nodes also allow redundancy in case of one node failure.

c) **Aggregation Tasks Supervising and Controlling** – Example: query management and distribution tasks are placed on an additional separate node in the same cloud. This allows additional scalability to support a larger user basis from all the projects residing on the same cloud. In the case of one project having hundreds of users, a dedicated aggregation node is then assigned to such a project not to influence other projects residing on the same cloud. The fast query management process is not impacted by the other activities as described above.

How the Inzata Core MPP Engine works

This section contains a detailed description of principle technology features of the Inzata core aggregation engine including a description of how the automated reports generation works:

A) The source data is stored in a set of data clusters (data mesh), each of them representing a specific multidimensional data space. From the physical data model perspective, each data cluster contains a set of columns. Data is organized in a columnar form physically, i.e. data is vertically partitioned. Additionally, each column is partitioned horizontally, so one data cluster contains a mesh of data chunks. This approach allows for the data to be processed in a fully distributed environment (computer cloud). Data chunks are physically stored as standard files using a standard computer filesystem.

B) Each of the data chunks can be presented in several versions simultaneously, enabling a concurrent execution of queries against different versions of the data repository. Because the repository is partitioned both horizontally and vertically, there is only a need for copying really changed data chunks. This approach provides a very efficient repository space occupation, good cache utilization, and low latency for the creation of a new data version. Additionally, it enables the ability to load a new version of data to the repository without interruption of concurrent reporting services while ensuring atomic switch to the new data version.

C) From the user perspective the multidimensional data space is described by a logical data model, which uses only two classes of objects: facts and attributes. Facts represent measures on the atomic dimensional level of each available multidimensional data space (that corresponds to the above-mentioned data cluster on the physical level). Attributes are sets of elements, along which the facts are sorted. Attributes can be organized in hierarchies. Each of such hierarchies, connected to a specific fact (or set of facts), represents one dimension of a multidimensional data space. The number of dimensions is virtually unlimited (limited mostly only by the space of physical data repository in a computer cloud). The above-mentioned approach to physical data storage architecture ensures there is no impact of the total number of facts, attributes, and dimensions to the performance of the certain query (that is affected by actually used objects only).

D) A new special Analytical Query Language ("AQL") has been designed for querying of above-mentioned multidimensional data space. It uses logical model objects only, so the AQL query is completely independent of the physical data model and the state of the physical data storage (unlike other solutions based on materialized multidimensional cubes). AQL is able to describe an unlimited number of nested multidimensional data space projections.

E) An automated query tree generator provides Logical Query decomposition accordingly to the Logical Data model of the multidimensional data space. The result of such projection defines a new multidimensional space that can be used as a source for any future projections. Each of such multidimensional spaces can be projected to each other along the common dimensions. The queries can be organized to a virtually unlimited acyclic directed graph. The subgraph of such graph connected to one root node is technically called the Query Tree. This query tree can be optimized (pruned) using the rules for the union of the projection operations. The root node of such a tree represents the multidimensional report result. Described decomposition approach in conjunction with above mentioned physical data storage architecture enables efficient caching of already projected multidimensional data spaces. This way the physical data storage is dynamically expanded and allows subsequent run-time optimization of future queries. There is also a possibility to materialize some multidimensional subspaces in advance to optimize the efficiency of execution on huge sets of similar reports in batch mode.

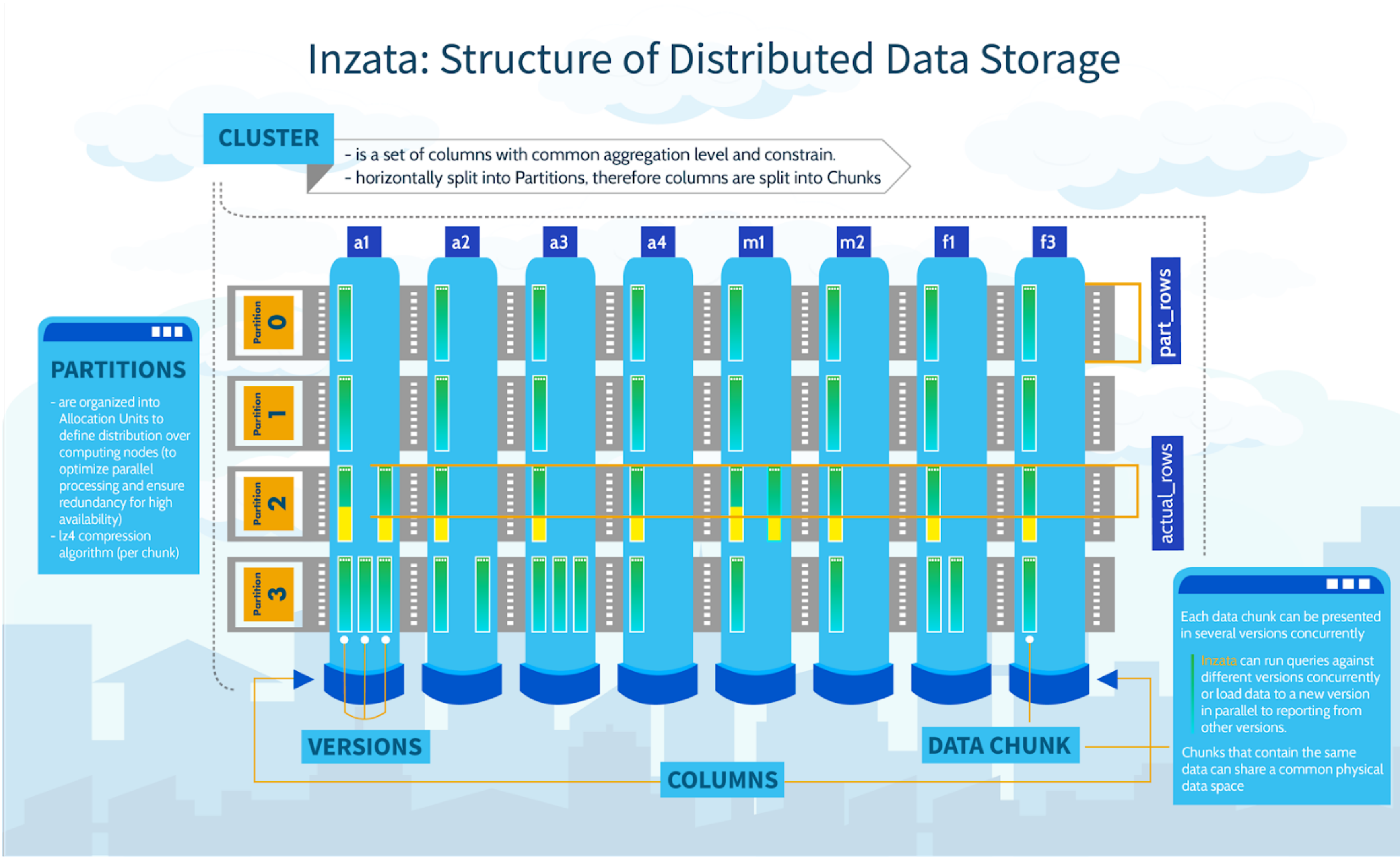
F) Each of the nodes of the above-mentioned Query Tree is transformed into a Macro Language, which represents the physical execution plan of the particular multidimensional data space projection. The Macro code is parallelized to be able to effectively utilize distributed (cloud) computing systems. The parallelized macro language is then distributed to task queues of the desired cloud nodes, where is subsequently transformed into the low-level execution code.

G) Concept of the physical task execution is based on Map Reduce programming model to allow massive parallel execution with good scalability. There are several basic operations, needed to ensure the above-mentioned projections of the multidimensional data space:

1. Selection of the source data accordingly to the specified constrain
2. Arithmetic operations and functions with the data on the atomic level of the source multidimensional data space
3. Change of the dimensionality of the atomic level data to be able to project them to the target multidimensional data space

Points 1. and 2. typically fall to the Map phase and covers a number of basic operations such as compression of attribute keys, generation of constraining bitmaps, filtering of the source records, data typecasting, and arithmetic operations with source data on the level of original data space dimensionality.

Point 3. is accomplished during the Reduce phase. The operation is a grouping of the fact data accordingly to values of the keys with the application of the aggregation function (such e.g. SUM, MIN, MAX). The specific algorithm has been designed with respect to CPU cache utilization, asymptotic complexity of the algorithm, and ability to massive parallel execution. Each reduction level ensures sorted order of resulting keys, so a fast sort-merge algorithm can be used on any level of reduction (even on macro-level to merge results of data chunks between distinct physical cloud nodes). The described approach is very suitable for use in a computing environment containing massive parallel processors because parameters of the reduced process (e.g. number of concurrent threads, the volume of data processed per thread, etc.) can be optimized accordingly to the characteristics of both computing systems and processed data. This way the process can be executed in a non-uniform hardware cloud as well.



Inzata: Structure of Distributed Data Storage
CLUSTER
- is a set of columns with common aggregation level and constrain.
- horizontally split into Partitions, therefore columns are split into Chunks
a1
a2
a3
a4
m1
m2
f1
f3
Partition 0
Partition 1
Partition 2
Partition 3
part_rows
actual_rows
PARTITIONS
- are organized into Allocation Units to define distribution over computing nodes (to optimize parallel processing and ensure redundancy for high availability)
- lz4 compression algorithm (per chunk)
VERSIONS
COLUMNS
DATA CHUNK
Each data chunk can be presented in several versions concurrently
Inzata can run queries against different versions concurrently or load data to a new version in parallel to reporting from other versions.
Chunks that contain the same data can share a common physical data space

How can Users benefit from the Inzata Analytical Platform?

As the first technology of its kind, Inzata Engine brings enterprise analytics based on massive parallel processing to a much more affordable level. Till now, similar technologies have been used in enterprise projects by very expensive vendors (Teradata, Vertica, MicroStrategy, IBM Netezza, etc.). Enterprise BI requires analysis at all levels of the business model (Requirement for atomic-level analysis).

There are a variety of technical factors that differentiate the Inzata Analytics Platform in the marketplace. We want to differentiate for the Banking functionality, and the functionality of the enterprise's BI architectures are measured by *Data depth* -- access to complete corporate data stores including detailed data --- because much of the time that is where the answer to the problem lies, especially for applications in Banking, for example. *Data breadth* -- can the system support sophisticated industry business models and diverse applications? With data breadth, you can ask all the questions of the decision support system that you are going to want to find the answer to your business problem. The Inzata Platform with its Inzata Engine represents the absolute top BI technology because we offer the most functionality -- full support of data depth, data breadth, and reporting range. Some competitors in the market today have fair depth support, but they rely on a star schema so you are not going to get much breadth, and to get the full depth you may need support for differently keyed fact tables. That's a data breadth problem. They also require one lookup per dimension. A problem for sophisticated business models requiring characteristic analysis, cross-dimensional attributes, etc.

Summary analysis can be valuable. However, this is typically where you find the problem. Some costs at a certain aggregate level are extremely high, certain units are not as profitable as you'd expect. Now, many tools can handle this summarized data. They can help you find the answer to your problems. However, to find the underlying cause you will require detailed analysis at the atomic transaction data level to understand what is causing such high costs, and why plans are not performing well. It has been said that the devil is in the details. Our Inzata tools are designed to help you find those problems and correct them in real-time. Essentially, you will need claim or lab level analysis (equivalent to the transaction) to find the root cause of the problem, to find over-utilization, you must go to the claim level and find which procedures, prescriptions are being offered more often, for a higher cost than you expected. You will want to do a summary and detailed analysis, but the real value of data can be found at its transaction level. Transaction level analysis gives you the ability to do for example affinity analysis. In Banking, the example could be: What additional products are used by profitable customers with high checking account balances? The reason the Inzata Engine can do this is that we have a massive parallel engine powering our platform. Transaction level analysis requires a platform engine with very robust and powerful analytical query processing.

Banking

In **Banking**, BI requires analysis at all levels of the business, especially at the detail level. Summary analysis -- at the Region, state, and branch levels -- can be valuable. However, this is typically where you find problems. It has been said that the devil is in the details. Our Inzata Analytical platform is designed to help you find those problems and correct them in real-time. Essentially, you will need transaction-level analysis to offer the right products, to the most profitable customers (and less risky customers), at the right time. You will want to do a summary and detailed analysis, but the real value of a data warehouse can be found at the transaction level. Transaction level analysis gives you the ability to do market basket analysis. The market basket is a very hot trend in Banking. It allows you to find product affinity. For example, a profitability analysis question might be: What additional products are used by profitable customers with high checking account balances. By understanding your most profitable customers' account portfolio, you can cross-sell additional products and services to maximize your key assets -- your customers. With product affinity analysis, you can also design better promotions. You can become the one-stop financial services provider for desirable customers, and transaction-level analysis will help you to achieve this goal. Another question like this cannot be answered by multidimensional cube technology tools :

- For customer segments with > 1,000 members in the current month, rank the top 10 customers based on profitability. Show the profitability, number of customers, and total billed and paid amounts over the prior 12 months.

Non-aggregable metrics are very important to Banking, as well as in Retail, you cannot sum inventory over time. If you did, you would have a lot more product on-hand than was ever shipped. Some examples of non-aggregation metrics for banking are month-end balance, risk score, profit score, and interest period.

INSURANCE

Enterprise Insurance BI requires an extensive reporting range of broad, deep data. This means that you need analytic power to do multi-level metrics such as contribution, market share, and gross margin, just some examples. Multi-level metrics occur when information is calculated from a different level in the dimensional model. For example, the contribution of a product to branch sales. Conditional metrics such as channel sales $, # of atm customers, and branch variance are also key to solving business problems. Conditional metrics occur when the calculation is given a specific condition such as $ only for certain channels, Variance for a particular branch or branches. The analytical engine must be able to constrain the calculation in an intelligent fashion, otherwise, your calculations and totals could be incorrect. This intelligence is built into Inzata Engine and its module Query Tree Generator.

Time-series analysis is very important to the business user, it is key to spotting trends and capitalizing on those trends. This year/last year is key to understanding product performance at yearly intervals. Therefore, for real-world BI analysis, the tool (Inzata Platform) must support the list of analytics, multi-level metrics, conditional metrics, transformation metrics, qualifications, and ranking, these are the types of analysis that are going to be important in solving business problems. Business users need to analyze data for such things as: How are sales trending? Where are profits headed? Is our advertising campaign working? Business users need this information to perform their jobs optimally. You can also perform impact analysis as your business rules change. If, for instance, you change your definition for a metric such as profitability, you can measure how this will impact all the reports, templates, throughout the system. Examples of Insurance business questions supportable by Inzata are:

- Who were our 100 most profitable agents last year? For each of those 100 agents, which 2 products were their most profitable?
- How many insurance claims were settled within two weeks of the accident?
- What is the total transaction amount for straight payments by unit and transaction date for 7/1/18 through 7/31/18, where the date of loss is between 6/27/18 and 6/30/18?
- What was the average amount of time that passed between the accident and first contact? First contact to claim payment?

With Inzata, the model can be optimized based on your business rules, the types of questions you ask, and your assumptions about your business. With Inzata's enterprise solution and data integration approach, a user has the ability to support many characteristics of their customer such as age, income, gender, if they own a car. If you can find the demographic profile of your most profitable customers, then find the products that are most appealing to these groups, then you can develop target marketing campaigns, design new products, and services, etc. Transaction level analysis becomes essential to data breadth for market basket analysis. Also, data breadth is required to support customer demographic analysis (characteristics of your customers), if you want to integrate external data to try and understand what kind of customer is coming in and using your financial services. You are going to need data breadth support to get those characteristics in your model.

Healthcare

There are cube vendors and other query tools trying to provide drill through and drill around to try and get to deeper data and try to provide better answers for product and customer analysis and so forth, but it's not a real solution, only a patch. For example, such **healthcare** questions cannot be answered by multidimensional tools like Power BI:

- Show the total cost and percent of all medical costs for diabetics,
- Show total cost and percent of all medical cost for at-risk diabetics based upon individual patient lab results,
- Show me by in and out of network, the number of claims, billed amount, and paid amount for groups with members that have children, both work, are over 35 and have,
- Show me the top 100 members ranked by prescription volume.
- How many used multiple pharmacies, refilled a prescription too early, and have ever obtained a prescription for a DEA controlled substance used an emergency room service at a non-participating hospital?

These technologies are old, they were built 20 years ago when the understanding of data size and breadth had not come to the forefront. The Inzata architecture has been built in the modern era of non-relational databases when we understood that you needed a complete platform with a very powerful, massive parallel with a sophisticated and universal business layer. The Inzata Query Generation Engine is designed to provide answers against large amounts of data by directly accessing (and leveraging) the power of the Inzata Engine and in such a way that allows you to analyze data depth, breadth, and ad-hoc reporting range is ahead of the current market.

DBA Support

With the Inzata Analytical platform, we also provide the equivalent for your DBA -- different statistics to show BI tool usage, maintenance, ROI. The DBA wants to answer questions such as: Who is using the BI Platform? When? With what frequency? We provide user base and HW resources monitoring with the embedded Inzata Activity Monitoring tool and already pre-built analytical dashboards. Inzata also provides an admin environment to manage thousands of users and objects across multiple projects and their objects from the security and access point of view. You can create, delete, copy, distribute -- anything that you need to do from an administrative standpoint.

Conclusion

Business Intelligence Analysis of large data and a rapidly increasing number of users are representing challenges for business intelligence vendors as well as database engine vendors. Traditional hardware (HW) concepts, such as Row-based databases and multidimensional online analytical processing (OLAP) approaches utilizing current HW, do not provide sufficient speed and scalability to respond to current BI challenges, and thus a need exists.

Inzata's goal is to solve your organization's business problem, to help you understand your business better, and to leverage that valuable asset that you have in storage – your data – and in such a way as to monetize your data. We think your data is your asset -- you've made the deposits, now we want to be the key player in helping you get a full return on investment.