李玉衡

daniel.lee@mariadb.com

ColumnStore 产品测试和技术支持

灰色为遮挡区域，排版请注意

扫码观看大会视频

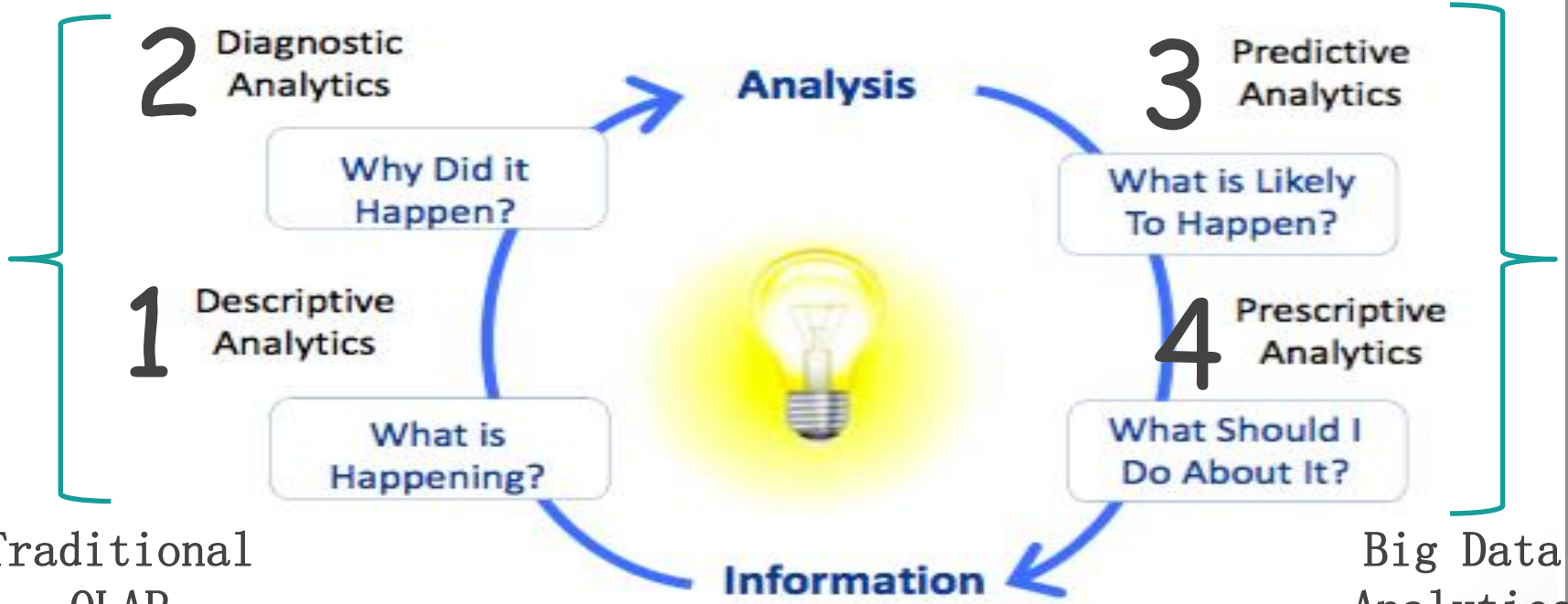
MariaDB ColumnStore Product Training

Content

- Analytics Introduction
- MariaDB Solution for Big Data Analytics
- MariaDB ColumnStore Deep Dive
- Use Cases and Differentiations
- Cassandra Compare
- Sizing and Pricing
- Target Audience Message

灰色为遮挡区域，排版请注意

Type of Analytics



Traditional OLAP

Big Data Analytics

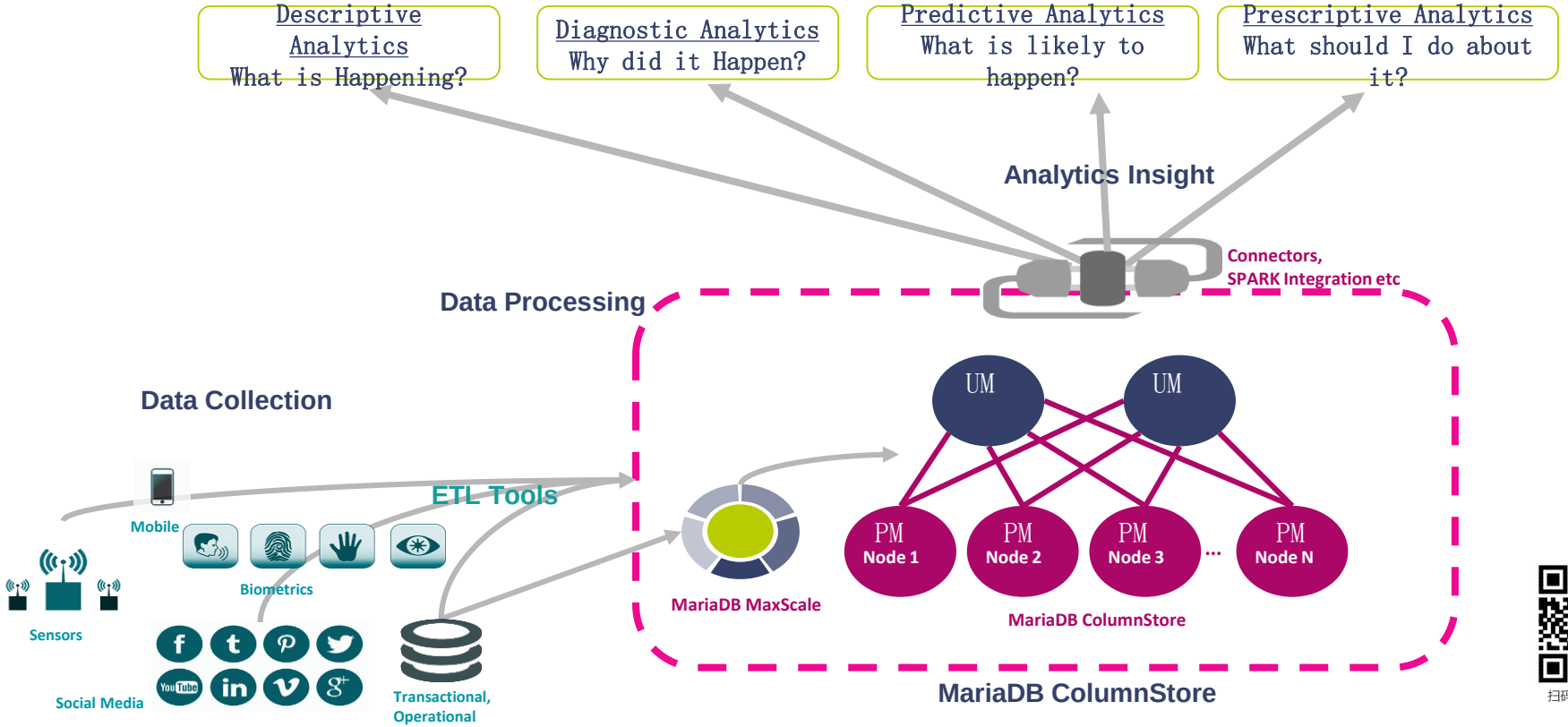
灰色为遮挡区域，排版请注意

Analytic Excellence Leads to Better Decisions

Gartner

MariaDB Solution for Big Data Analytics

High performance data management solution for big data analytics



灰色为遮挡区域，排版请注意

扫码观看大会视频

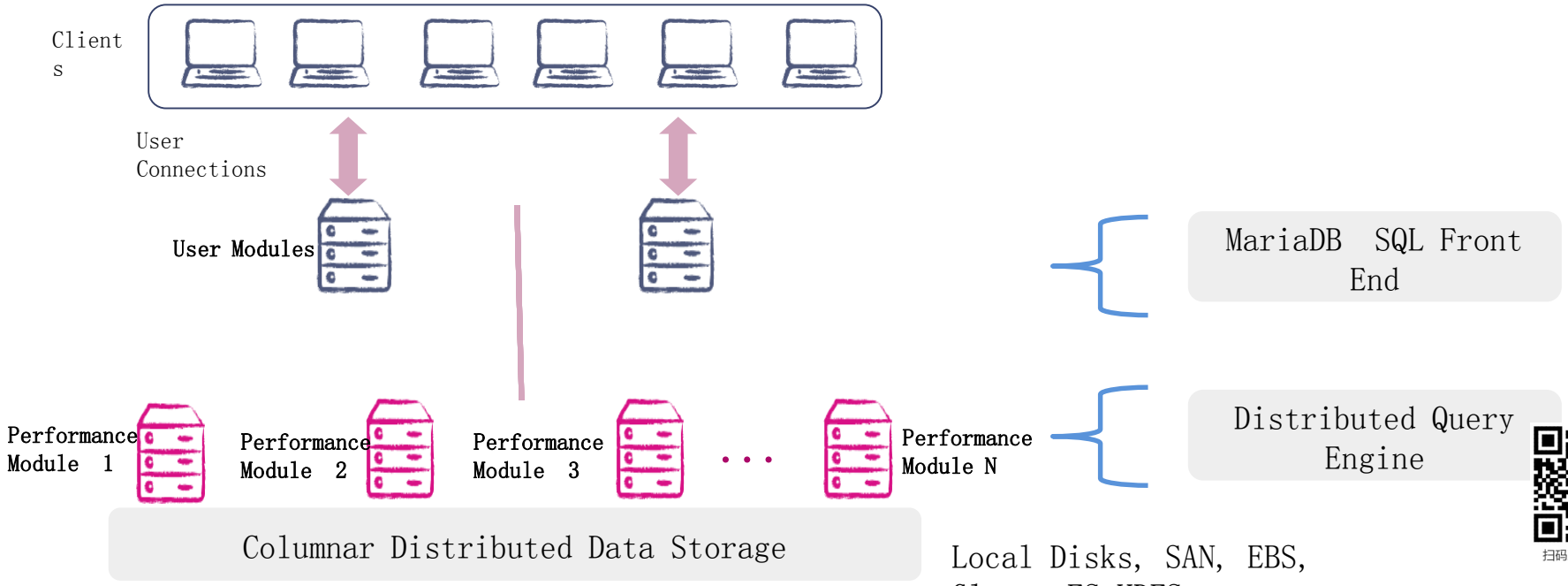
MariaDB ColumnStore Deep Dive

灰色为遮挡区域，排版请注意

扫码观看大会视频

MariaDB ColumnStore Architecture

- User Module : Processes SQL Requests
- Performance Module : Multi Threaded Distributed Processing Engine



灰色为遮挡区域，排版请注意

MariaDB ColumnStore 1.0

Performance	• Columnar Storage, multi-threaded and Massively Parallel distributed execution engine
High Availability	• Built in redundancy and high availability
Scale	• Linear scalability
Analytics	• In database analytics with Complex and Cross Engine JOINs • Windowing functions and UDFs • Out of box BI Tools connectivity, • Analytics integration with R
Ease of Use	• ANSI SQL compatible • ACID compliant • No indexes, No materialized views • No manual partitioning
Data Ingestion	• High speed parallel data load and extract • Create Table as Select, Like -- locally, cross database joins, or over ODBC
Security	• SSL support, Audit Plugin, Authentication Plugin, Role Based Access

灰色为遮挡区域，排版请注意

Client Access

- ODBC/JDBC
- MariaDB/MySQL Connectors
- BI tools

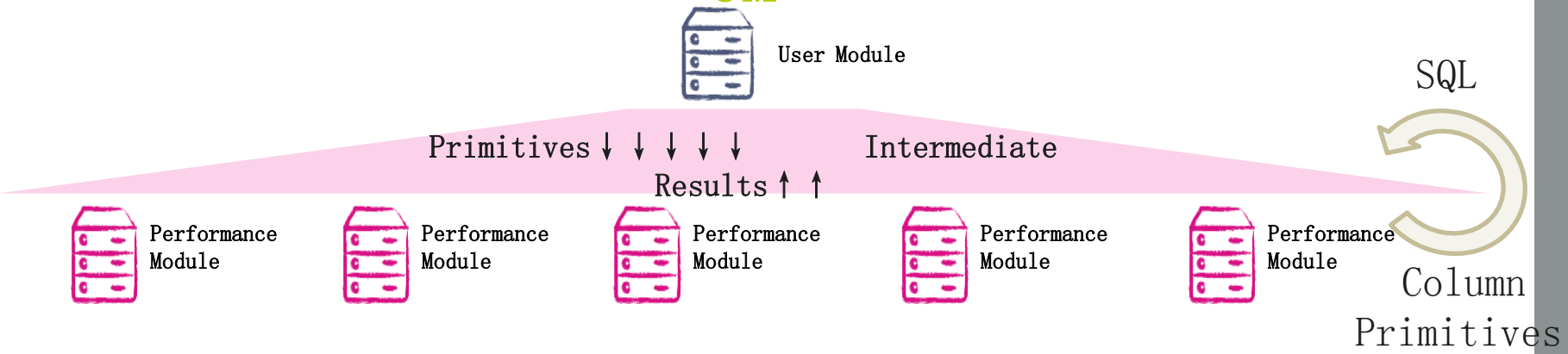
灰色为遮挡区域，排版请注意

Query Processing - UM

- Query parsed by mysqld on UM node
- Parsed query handed over to ExeMgr on UM node
- ExecMgr breaks down the query in primitive operations

灰色为遮挡区域，排版请注意

Query Processing – UM



SQL Operations are translated into thousands of Primitives

- Parallel/Distributed 2D Partitioned Data Access
- Parallel/Distributed Joins (Inner, Outer)
- Parallel/Distributed Sub-queries (From, Where, Select)

Query Processing – PM

- Primitives processed on PM
- One thread working on a range of rows
- Typically 1/2 million rows, stored in a few hundred blocks of data
- Execute all column operations required (restriction and projection)
- Execute any group by/aggregation against local data
- Return results to ExeMgr process in User Module
- Each primitive executes in a fraction of a second
- Primitives are run in parallel and fully distribute

灰色为遮挡区域，排版请注意

Query Processing – UM + PM

1. A request comes in through the Front end interface. MariaDB performs a table operation for all tables needed to fulfill the request and obtains the initial query execution plan from MariaDB Server.
2. Storage engine interface converts the MariaDB table objects to MariaDB ColumnStore objects. These objects are then sent to a User Module.
3. The User Module converts the MariaDB execution plan and optimizes these objects into an MariaDB ColumnStore execution plan. The User Module determines the steps needed to run the query and when they can run.
4. The User Module consults the Extent Map for the locations of the data needed to satisfy the query and performs extent elimination based on the information contained within the Extent Map.
5. The User Module sends commands to one or more Performance Modules to perform block I/O operations.
6. The Performance Module(s) carry out predicate filtering, join processing, initial aggregation of data, and sends data back to the User Module for final result set processing.
7. The User Module performs final result set aggregation and composes the final result set for the query

灰色为遮挡区域，排版请注意

Row-Oriented vs Column-Oriented

Row-oriented: rows stored sequentially in a file

Key	Fname	Lname	State	Zip	Phone	Age	Sales
1	Bugs	Bunny	NJ	11217	(123) 938-3235	34	100
2	Yosemite	Sam	CT	95389	(234) 375-6572	52	500
3	Daffy	Duck	IA	10013	(345) 227-1810	35	200
4	Elmer	Fudd	CT	04578	(456) 882-7323	43	10
5	Witch	Hazel	CT	01970	(567) 744-0991	57	250

Column-oriented: each column is stored in a separate file

Each column for a given row is at the same offset.

Key
1
2
3
4
5

Fname
Bugs
Yosemite
Daffy
Elmer
Witch

Lname
Bunny
Sam
Duck
Fudd
Hazel

State
NJ
CT
IA
CT
CT

Zip
11217
95389
10013
04578
01970

Phone
(123) 938-3235
(234) 375-6572
(345) 227-1810
(456) 882-7323
(567) 744-0991

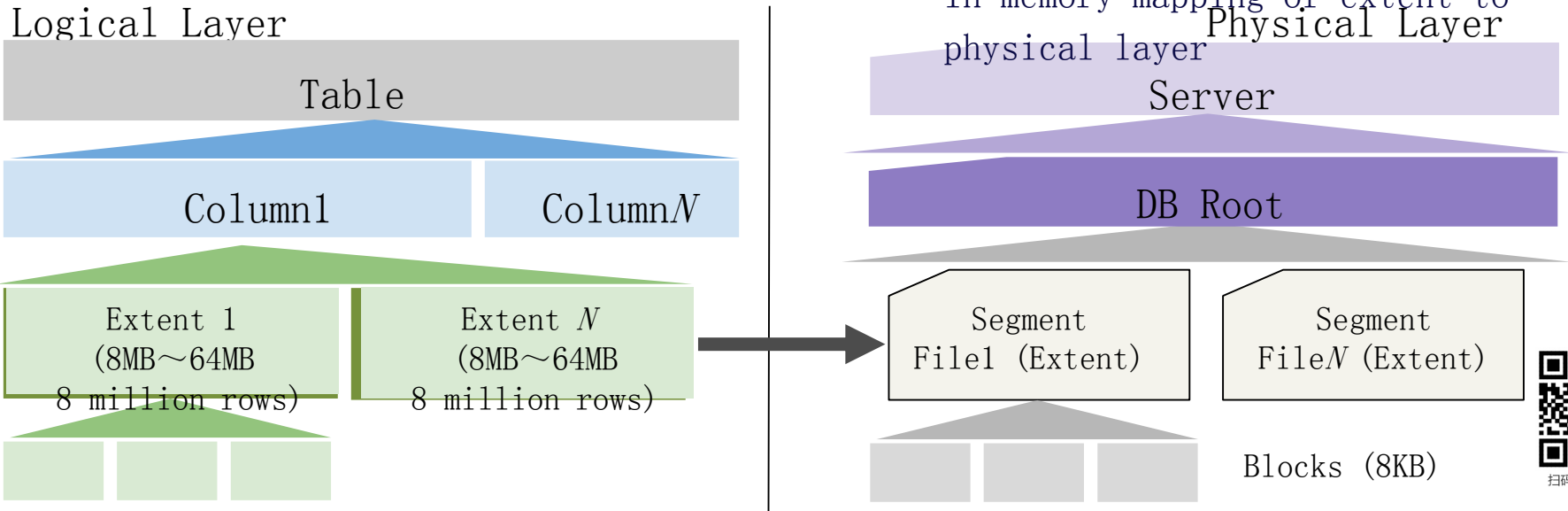
Age
34
52
35
43
57

Sales
100
500
200
10
250

灰色为遮挡区域，排版请注意

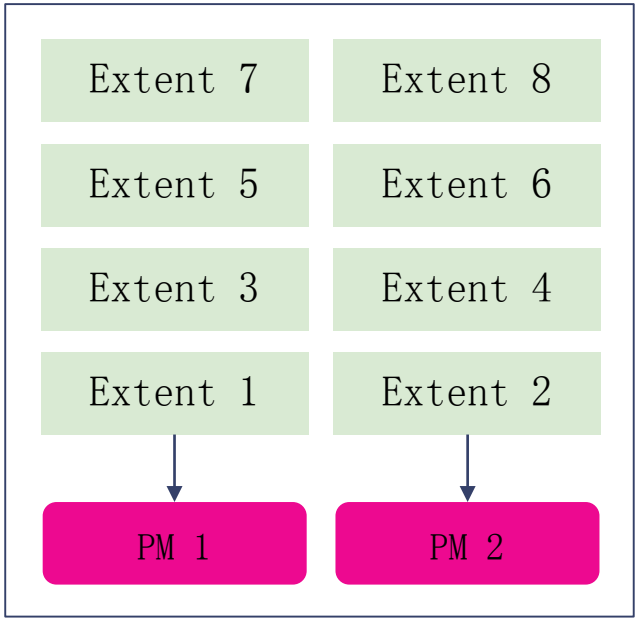
Data Storage

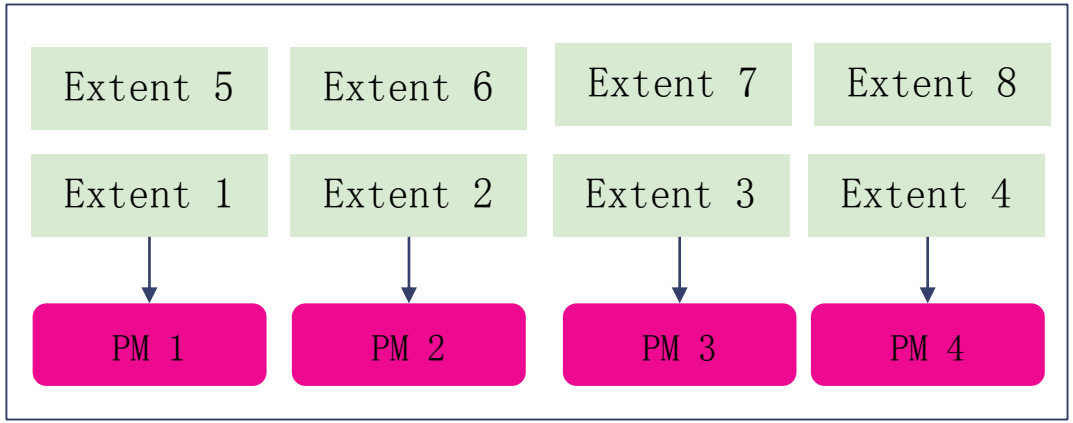
- Vertical Partitioning by Column
  - Each column in its own column file
  - Only do I/O for columns requested
- Horizontal Partitioning by range of rows
  - Logical grouping of 8 million rows of each column file
  - In-memory mapping of extent to physical layer



灰色为遮挡区域，排版请注意

Data Storage - Extents and PMs





灰色为遮挡区域，排版请注意

- Extent Map
  - In memory meta-data of an extent’s min, max value for a column, extent’s physical block offset and PM on which the extent resides

Data Storage - Local Disks

- Each PM nodes stores data on local disk
- No PM node can access the data on another PM node
- Shared Nothing
- No data redundancy

灰色为遮挡区域，排版请注意

Data Storage - SAN

- Each PM node is attached to a set of volumes on SAN - called DBRoots
- Upon failure of PM node, another PM attaches to the failed PM’s DBRoots
- Shared nothing during running state
- No data redundancy

灰色为遮挡区域，排版请注意

Data Storage - GlusterFS

- Distributed file system
- Software based storage system
  - GlusterFS runs on every PM node
  - Creates distributed file system with each PM node's local disks and network interface across PM nodes
- Data redundancy across multiple nodes
- Automatic data failover
- Data availability during failover and failback

灰色为遮挡区域，排版请注意

Data Storage - EBS

- Dynamic scaling to handle variable workloads
- Data layer high availability with Elastic Block Store (EBS)

灰色为遮挡区域，排版请注意

Data Ingestion

- Bulk data load
  - cpimport : CSV and Binary
  - LOAD DATA INFILE: CSV
- Apache Sqoop Integration:
  - Integration with cpimport and sql interface
- Future Release
  - Data Streaming from MariaDB/MySQL database to MariaDB ColumnStore cluster
    - via Kafka
    - Avro data record

灰色为遮挡区域，排版请注意

Data Ingestion - cpimport

- Fastest way to load data
  - Load data from CSV file
  - Load data from Standard Input
  - Load data from Binary Source file
- Multiple tables in can be loaded in parallel by launching multiple jobs
- Read queries continue without being blocked
- Successful cpimport is auto-committed
- In case of errors, entire load is rolled back

灰色为遮挡区域，排版请注意

Data Ingestion - LOAD DATA INFILE

- Traditional way of importing data into any MariaDB storage engine table
- Up to 2 times slower than cpimport for large size imports
- Either success or error operation can be rolled back

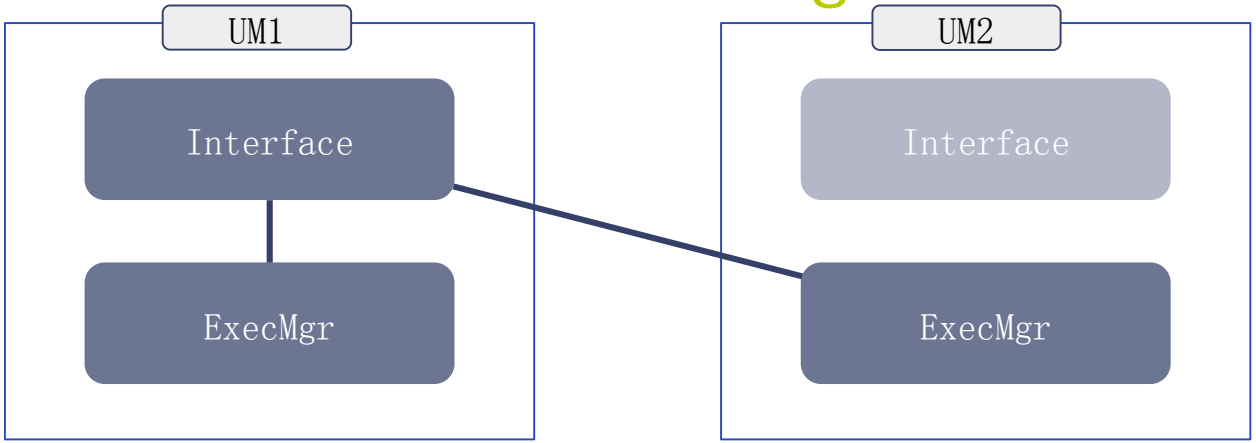
灰色为遮挡区域，排版请注意

High Availability

- HA at UM node
  - When one UM node goes down, another UM node takes over
- HA at PM node
  - SAN/AWS EBS - When a PM node goes down, the data volumes attached to the failed PM node gets attached to another PM
  - Local Disks -If a PM node goes down, the data on its disks are not available, though queries continue on the remaining data set
- HA at Data Storage
  - AWS EBS
  - GlusterFS- Multiple copy of data block across storage. If a disk on a PM node fails, another PM node will have access to the copy of the data
  - HDFS - Multiple copy of data block across storage. If a disk on a PM node fails,

灰色为遮挡区域，排版请注意

Multi-UM
Default configuration

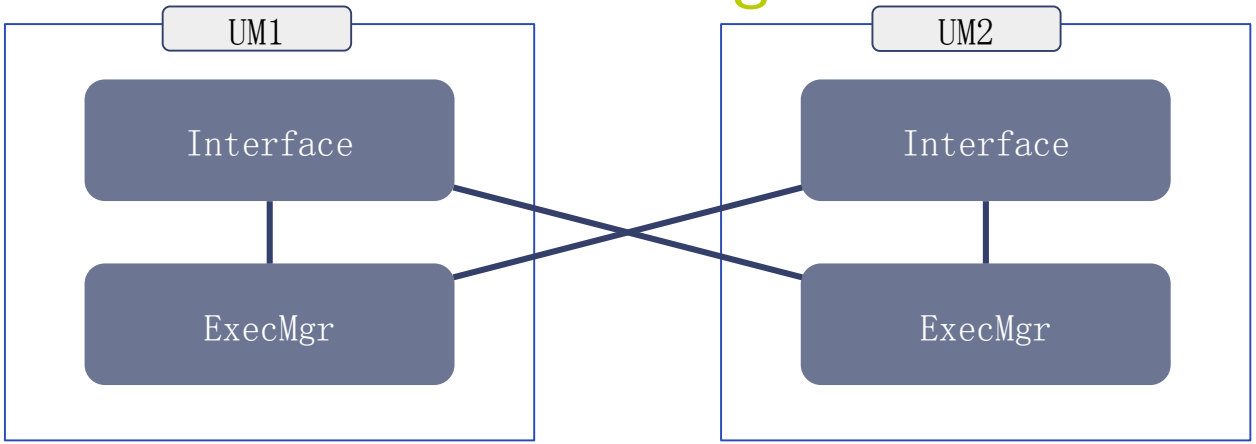


Connection Id based round-robin

- Applications connect to single UM
- Automatic round-robin distribution/scale-out of queries (based on connection id) across all UMs
- The two UM’s schema and non-ColumnStore tables to be kept in synch with mysql replication. Setup during post config or use enableMysqlReplication in mcs console

灰色为遮挡区域，排版请注意

Multi-UM Multi-Front end configuration



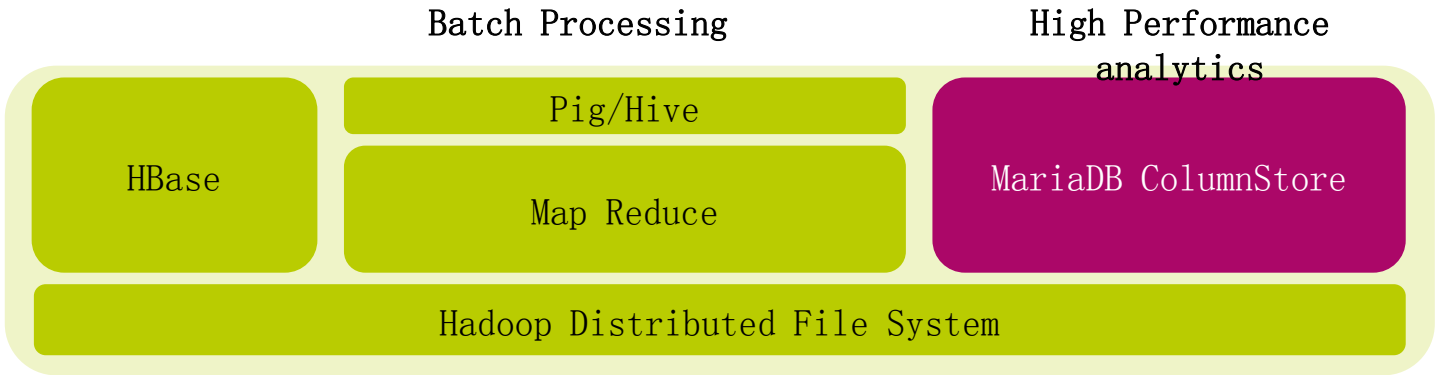
Connection Id based round-robin

- Applications can connect to multiple UM
- From each UM Automatic round-robin distribution/scale-out of queries (based on connection id) across all UMs
- The two UM's schema and non-InfiniDB tables to be kept in synch wi mysql replication - Setup during post config or use enableMysqlReplication in calPont console

灰色为遮挡区域，排版请注意

MariaDB ColumnStore on Hadoop

- Native scoop integration
- Runs on existing Apache Hadoop hardware
- SQL access to Apache Hadoop data
- libhdfs integration



灰色为遮挡区域，排版请注意

MariaDB ColumnStore on AWS

- Automated cluster installation on AWS
- Dynamic scaling to handle variable workloads
- Data layer high availability with Elastic Block Store (EBS)

灰色为遮挡区域，排版请注意

Use Cases

灰色为遮挡区域，排版请注意

Use Cases

PERFORMANCE AT SCALE

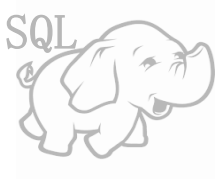
Put massive data sets to work with real-time analytics for your growing business

NEW INSIGHTS

Uncover new insights and business opportunities with advanced big data analytics

UNIFIED SIMPLICITY

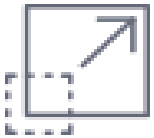
Simplify and reduce operational costs by uniting analytical and transactional workloads



HIGH PERFORMANCE ANALYTICS for HADOOP

Democratizes access to data in Hadoop to larger user base

Differentiators

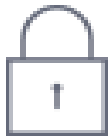


SCALE

- Massively parallel architecture designed for big data scaling to process petabytes of data
- Read performance scales linearly with data growth

SPEED

- Exceptional performance
- Real-time response to analytics queries



SECURITY and RELIABILITY

- Data with encryption for data in motion, role based access and audit features of MariaDB Enterprise
- Built-in high availability at access and data layers

SIMPLICITY with POWER

- Simplified management and maintenance, Easy installation and scaling
- Same interface as MariaDB and MySQL. Attaches to wide range of BI tools

灰色为遮挡区域，排版请注意

扫码观看大会视频

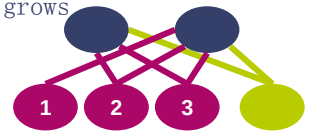
Use Case: Scaling Big Data Analytics

Business Challenge

- An organization is generating large amount of operational data
- Multiple tera-bytes of historical data
- With growth in business and in operational data
  - Analytics query performance degrades
  - Impractical to do analytics

MariaDB ColumnStore Solution

- Put past data into MariaDB ColumnStore
- As data grows



Add new node(s)

MariaDB ColumnStore 1.0

- Perform analytics without performance degradation
- Linear Scalability with data growth

1 100 10,000 1,000,000 100,000,000 10,000,000,000 100,000,000,000

10-100GB 100-1000GB 1-10TB 10-100TB...PB

Rows/DataSize Scope

MariaDB Enterprise OLTP

MariaDB Enterprise ColumnStore

- Harvest new value from large historical datasets by deriving new insights
- Support growth in your business, while continue to deliver high service

灰色为遮挡区域，排版请注意

Use Case: Discover Insight

As planes go through flights, various parts and engine of the planes need to be maintained for safety of the flights

- Analytics on the real-time data and historically collected flight parameter data
- Proactively project parts replacement, maintenance and air-plane retirement
- Too time-consuming to perform analytics with current toolset
- Most of the data analyst have SQL background

Real-time in-flight performance data

MariaDB ColumnStore Solution

Historical Data

Micro-batch upload real-time flight performance into MariaDB ColumnStore

- Complex-join, aggregation and windowing functions
- High speed real-time performance

Familiar SQL interface

Timely maintenance forecast, part replacement, flight retirement

The company plans to sell this solution as a service to commercial airliners

- Uncover new business opportunity with data exploration and analytics on big data

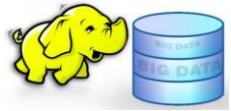
灰色为遮挡区域，排版请注意

扫码观看大会视频

Use Case: Accelerated Analytics with Hadoop

Business Challenge

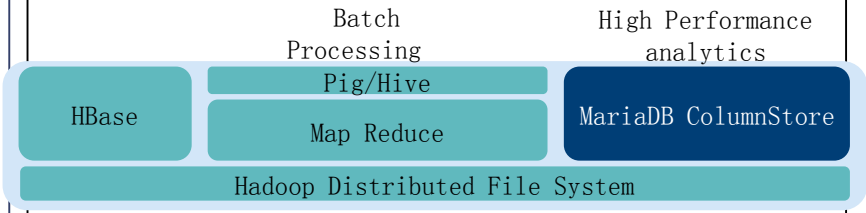
- Large amount of data in Hadoop



- Hadoop is suitable for
  - batch processing
  - Transforms via Map-Reduce programming
- Real-time analytics on Hadoop
  - Speed cannot meet business requirement with the Hadoop tool set
- Shortage of Hadoop skills for Data Scientist/
  - SQL in
  mature

MariaDB ColumnStore Solution

- MariaDB ColumnStore OLAP can run on premise, on cloud or on Hadoop cluster



- Ingest data from Hadoop
- Mature ANSI-SQL compliance
- Stellar performance : 70 to 80 times faster than SQL-on-Hadoop counterparts Hive, Hbase and Impala
- Mature interfaces

- Familiar SQL interfaces democratizes access to big data to larger user base
- Attach wide range of BI tools via MariaDB/MySQL connectors

灰色为遮挡区域，排版请注意

扫码观看大会视频

Use Case: Simplifying Big Data Management

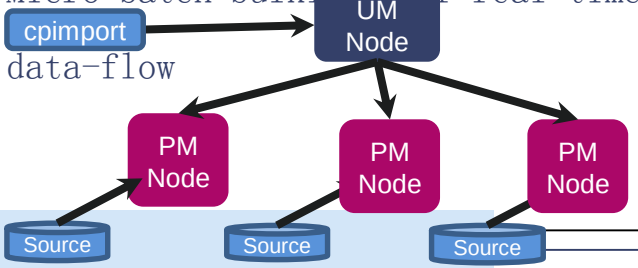
Business Challenge

- Complexity of data management increases as data volume grows
  - Tedious to keep up with indexes and partitioning as data grow
  - Scaling-out or Scaling up management
  - Moving operational data to big data analytics platform in real-time

MariaDB ColumnStore Solution

MariaDB ColumnStore

- Liberation from Index management
- Automatic partitioning
- Easy to grow
- Micro-batch bulkload for real-time data-flow



- Improved DBA productivity
- Reduced operational complexity
- Getting most value out of big data while minimizing DBA

Developer Friendliness

Developer Challenge

- Focus on application development rather than tuning queries and/or application as data grows
- Have flexibility to work varied tools and languages: SQL, BI tools, Python, Java, C++, Go
- Easily deploy and test analytics applications

MariaDB ColumnStore Solution

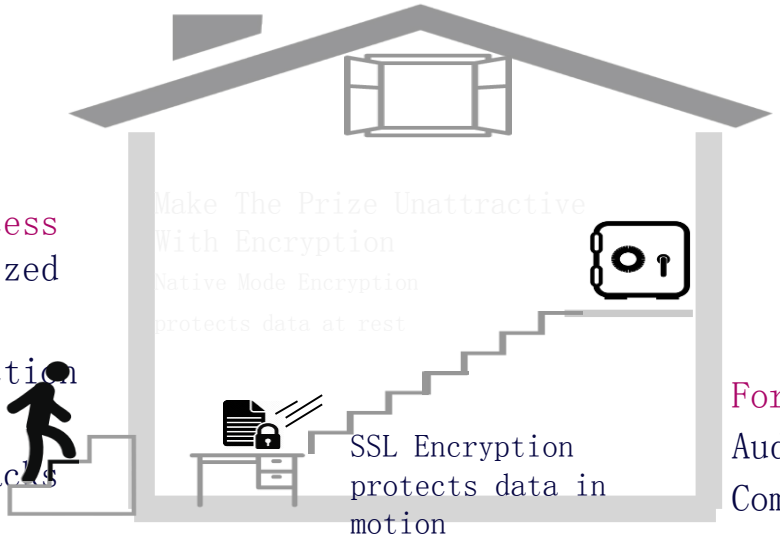
- MariaDB ColumnStore empowers developers with
  - No need to tune queries and applications as data grows
  - Mature SQL interfaces
  - Python, R, Java and C++ connector
  - BI tools access through ODBC/JDBC and MariaDB connectors
  - Cloud consumption options for AWS
  - Easy installation

- Improve developer productivity
- Leverage existing investments
- Minimize Opex

灰色为遮挡区域，排版请注意

MariaDB ColumnStore OLAP Security

- Built upon MariaDB Server 10.1 – secure open source database



Control Access
- Unauthorized Access
- SQL Injection attacks
- DDoS attacks

Community Improves Security

Forensics
Audit Log for Compliancy

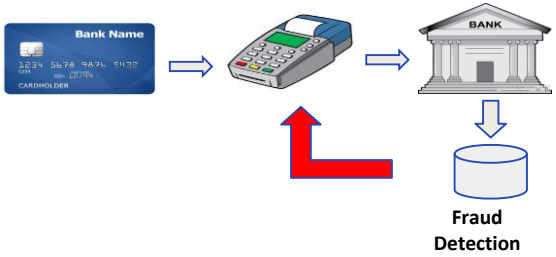
- Keep valuable data secure, while getting the most value out of your data assets
- Reduce Risks and costs associated with security breaches

灰色为遮挡区域，排版请注意

MariaDB ColumnStore High Availability

Business Challenge

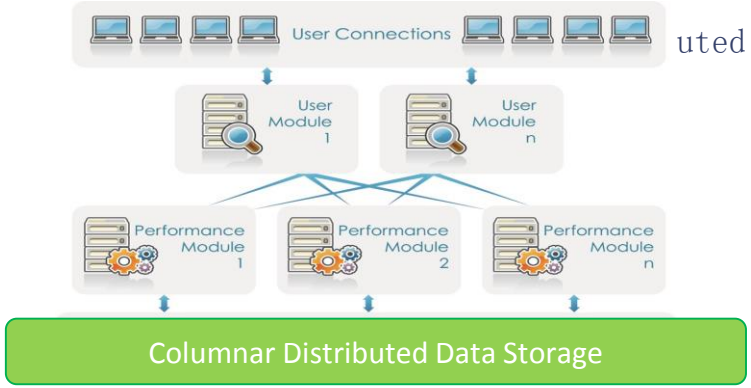
- A financial organization has mandate to detect fraudulent activities



- 2015 US total credit-card fraud cost $600 billion
- Each fraud incident average cost $1900
- Average 13 frauds per minute
- Any downtime in th system is costly

MariaDB ColumnStore Solution

- MariaDB ColumnStore ' s distributed, MPP architecture has built in high availability
  - Active/Standby data access nodes(UM)



uted

- Keep business running
- Minimize costs associated with downtime

灰色为遮挡区域，排版请注意

扫码观看大会视频

MariaDB ColumnStore: Performance Comparison

Performance for 1gb DTB3 database (in seconds)

	InnoDB	ColumnStore	Delta
cpimport	n/a	27.70	n/a
LDI	1,231.07	68.27	1,803%
InsertSelect	1,532.29	94.10	1,628%
DBT3 (disk)	3,881.40	21.07	18,421%
DBT3 (cached)	3,637.49	14.74	24,677%

Tested on Amazon AWS
Instance type: m4.2xlarge
Disk: SSD 200 GB without encryption, internal
Source data: On a 200gb EBS, attached

灰色为遮挡区域，排版请注意

2016 The Computing Conference

THANKS