AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

aws INNOVATE
ONLINE CONFERENCE

分会场六：大数据分析

利用 Amazon Redshift 搭建现代化的数据仓库

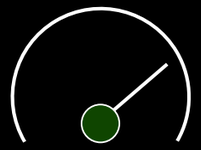
王晓野，AWS 解决方案架构师

Amazon Redshift – 数据仓库

完全托管的 PB 级数据仓库

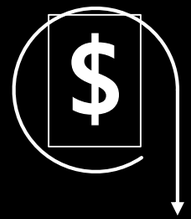
大规模并行架构，petabyte 级别扩展

性能优越



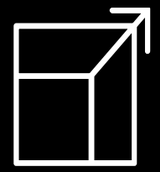
列式存储提高 I/O 效率
大规模并行查询架构

价格低廉



低至每 TB 数据 $1000 每年
传统数据仓库解决方案 1/10
的成本

轻松扩展



根据实际容量及性能需求
扩展或收缩集群

安全可靠

静态数据加密
Amazon VPC 隔离环境部署集群
AWS KMS 管理加密密钥

议程

Amazon Redshift 架构及核心概念

数据的存储和导入

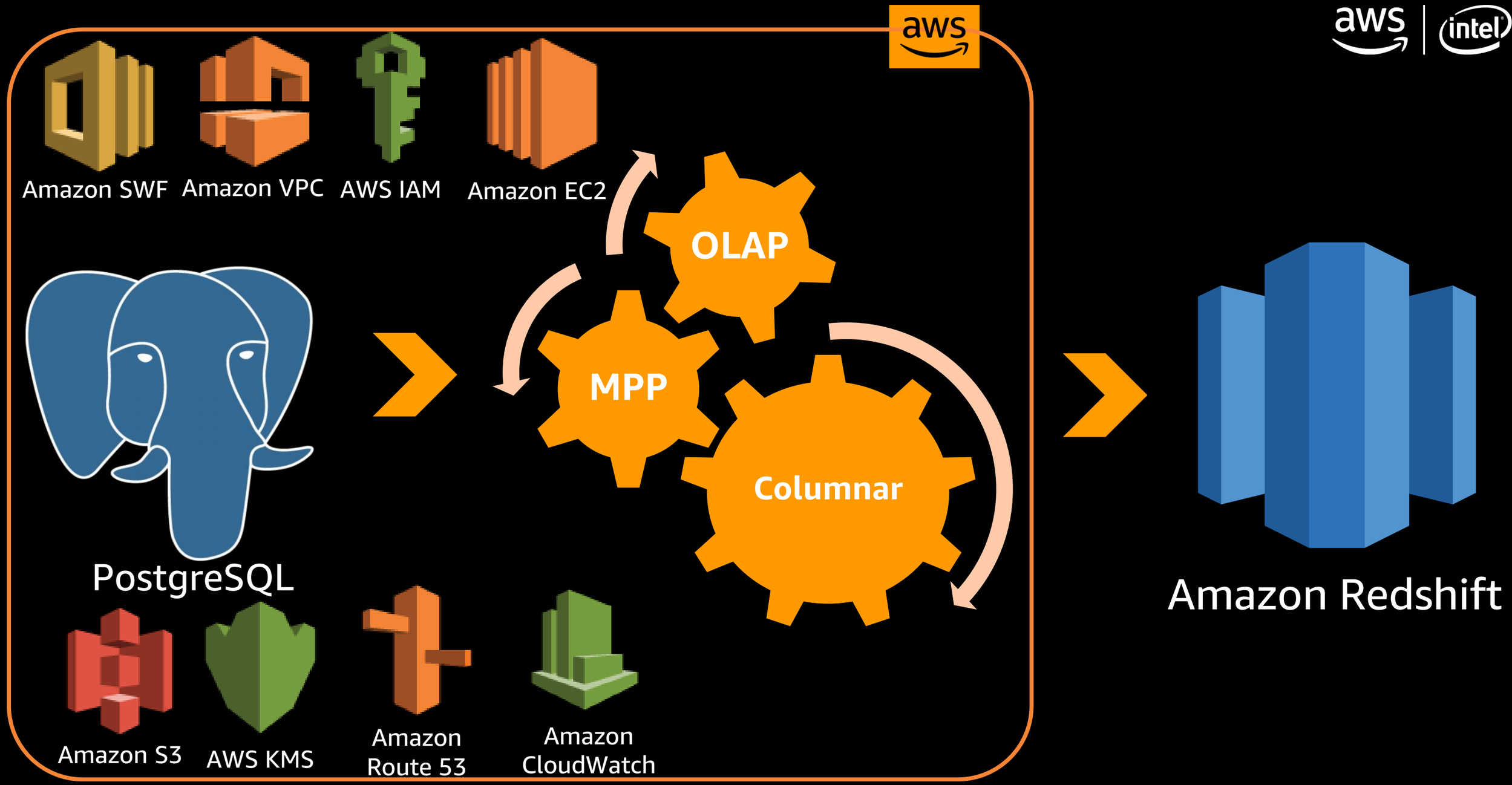
Amazon Redshift 的崭新特性

构建现代化的数据分析链条

架构及核心概念

aws INNOVATE
ONLINE CONFERENCE

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营



AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

2013 年 2 月

> 140 个重要补丁

> 220 项重要功能

2018 年 11 月

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

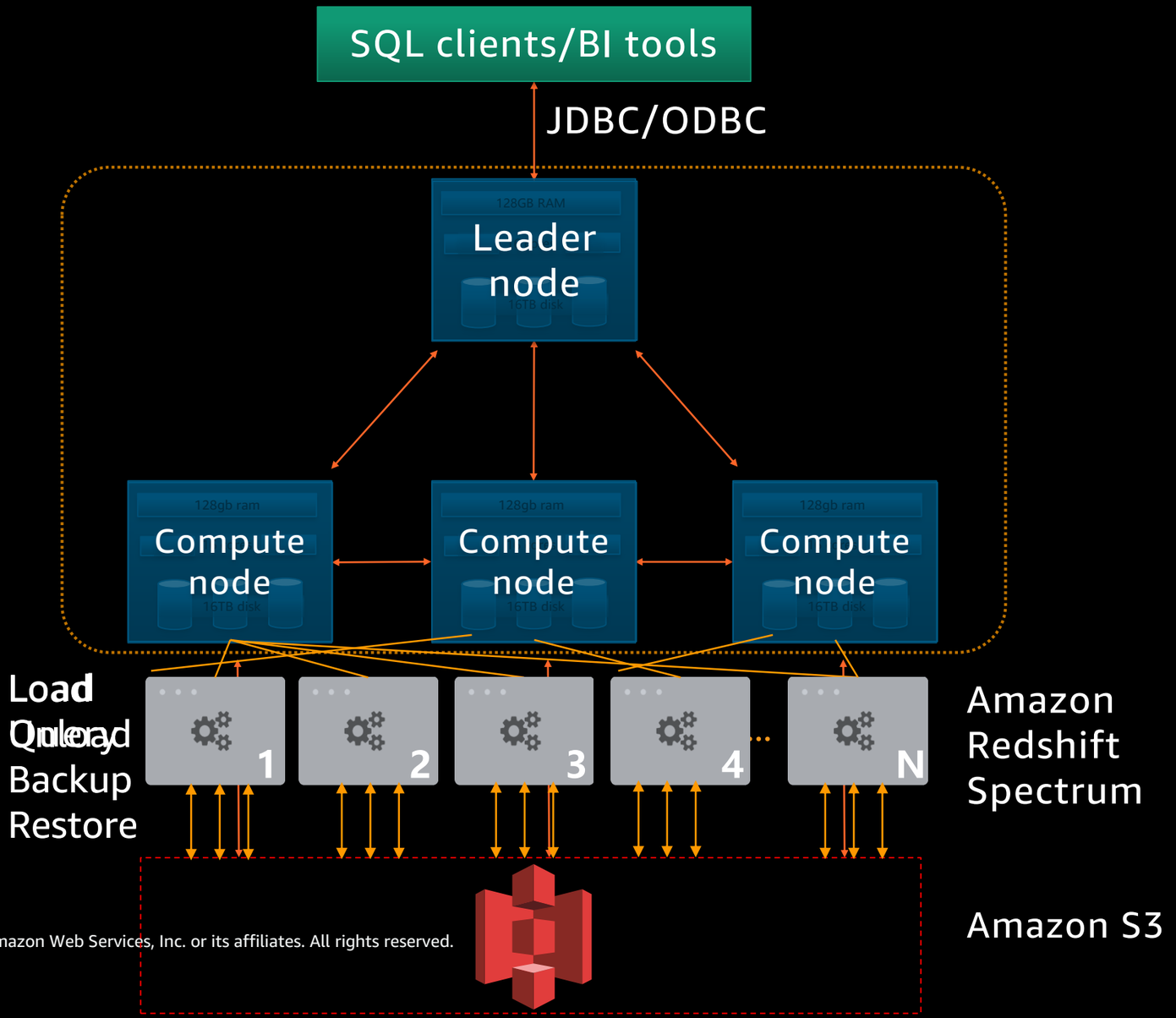
Amazon Redshift 架构

MPP 大规模并行计算，shared nothing
列示存储

Leader node **管理节点**
- SQL 接入点
- 存储数据库元数据
- 协调各个计算节点查询任务执行

Compute nodes **计算节点**
- 本地列式存储
- 并行化/分布式执行查询
- 数据加载、导出、 备份、恢复

Amazon Redshift Spectrum 节点
- 执行通过 SQL 直接对 Amazon Simple Storage Service (Amazon S3) 进行的查询



术语和概念 : Columnar 列式存储

Amazon Redshift 采用列示结构磁盘上存储数据

目的：减少分析查询的 I / O

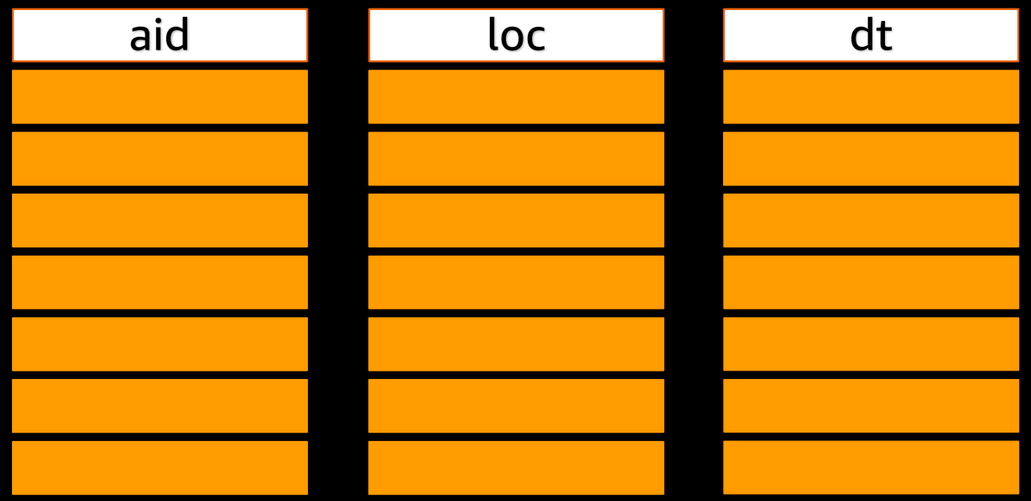
按列而不是行将数据在磁盘上物理存储

仅读取所需的列数据

Columnar 列式存储 示例

```
CREATE TABLE deep_dive (
    aid         INT             --audience_id
   ,loc         CHAR(3)         --location
   ,dt          DATE            --date
);
```

aid	loc	dt
1	SFO	2017-10-20
2	JFK	2017-10-20
3	SFO	2017-04-01
4	JFK	2017-05-14




```
SELECT min(dt) FROM deep_dive;
```

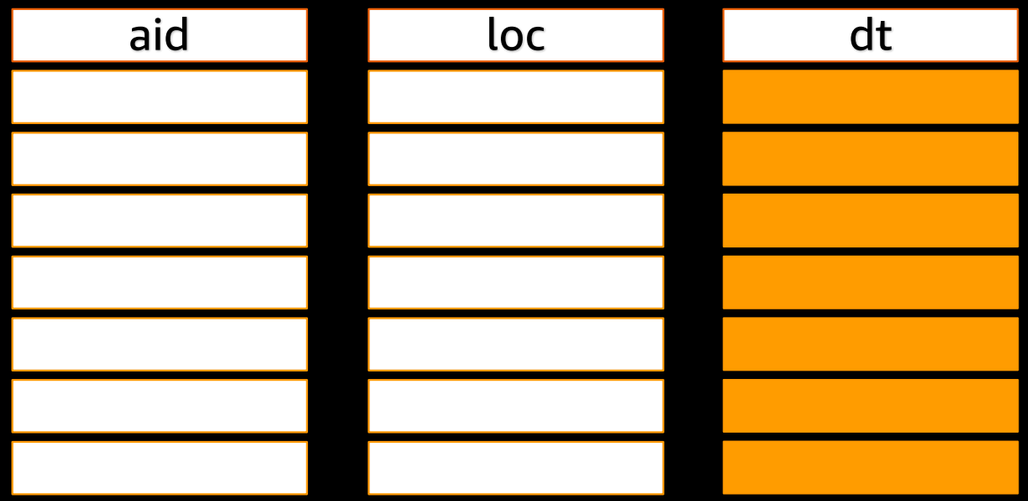
行式存储

- 需要将所有数据扫描
- 不必要的 I/O

Columnar 列式存储 示例

```
CREATE TABLE deep_dive (
    aid         INT             --audience_id
   ,loc         CHAR(3)         --location
   ,dt          DATE            --date
);
```

aid	loc	dt
1	SFO	2017-10-20
2	JFK	2017-10-20
3	SFO	2017-04-01
4	JFK	2017-05-14




```
SELECT min(dt) FROM deep_dive;
```

列式存储

- 仅扫描相关列的数据块

术语和概念 : Compression 压缩

目的:

- 允许在 Amazon Redshift 群集中存储更多数据
- 通过减少 I / O 来提高查询性能

影响:

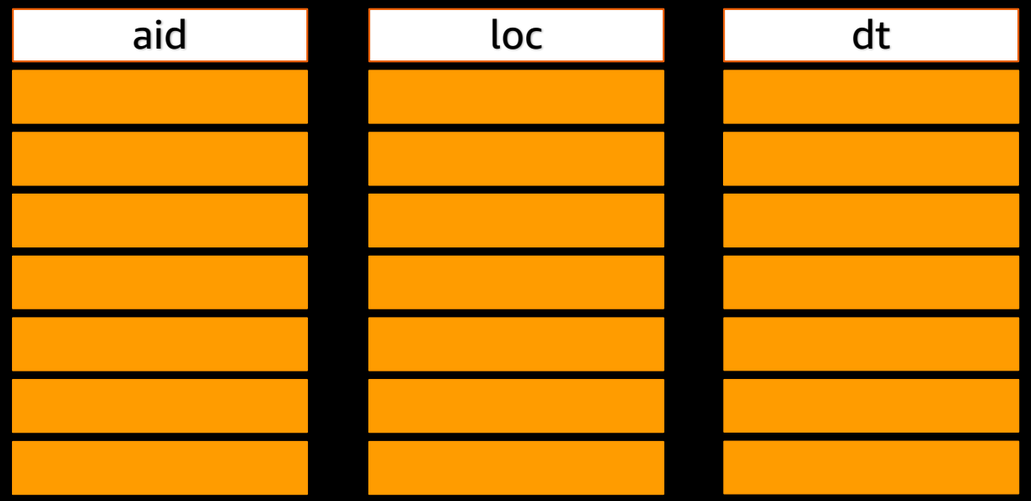
- 允许在群集中存储两到四倍的数据

默认情况下，COPY 命令会在第一次向空表中加载数据的时候自动分析，并将数据自动进行压缩

ANALYZE COMPRESSION 是一个内置命令，可以为现有表上的每个列找到最佳压缩编码格式

数据压缩示例

```
CREATE TABLE deep_dive (
    aid         INT             --audience_id
   ,loc         CHAR(3)         --location
   ,dt          DATE            --date
);
```

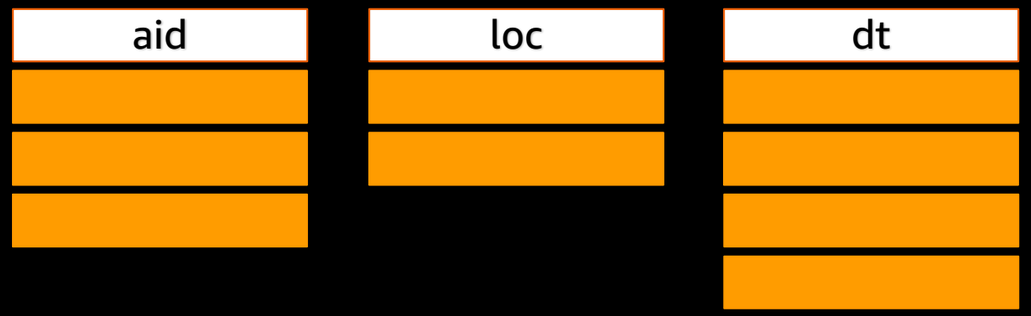


aid	loc	dt
1	SFO	2017-10-20
2	JFK	2017-10-20
3	SFO	2017-04-01
4	JFK	2017-05-14

为每列选择 11 种不同编码中的 1 种

数据压缩示例

```
CREATE TABLE deep_dive (
    aid         INT          ENCODE ZSTD
   ,loc         CHAR(3)      ENCODE BYTEDICT
   ,dt          DATE         ENCODE
RUNLENGTH
```



aid	loc	dt
1	SFO	2017-10-20
2	JFK	2017-10-20
3	SFO	2017-04-01
4	JFK	2017-05-14

由于采用列式存储结构，相同列中存储的数据有着相同的数据类型，因此能够过更有效的压缩

每一列数据的增长和压缩都与其它列独立

降低存储要求

减少 I / O

术语和概念：Blocks 数据块

列数据持久保存为 1 MB 大小的不可变块

数据块使用 12 种编码中的 1 种进行单独编码

完整数据块可能包含数百万个值

术语和概念 : Zone maps

目的 ： 避免不必要的 I/O

In-memory 数据块元数据

- 包含每个数据块的最小值和最大值
- 所有数据块都自动拥有 zone maps
- 为给定查询有效过滤掉不可能包含数据的数据块

术语和概念：数据排序

目的：通过提高 zone maps 的有效性和减少 I / O，使查询运行得更快

影响： 使得数据范围更合理的被限制在数据块中，结合 zone maps 元数据大幅缩小查询扫描数据范围

通过 SORTKEY 在表属性中定义一个或多列作为排序键来实现

最佳排序键取决于：

- 查询模式
- 业务需求
- 数据的属性

排序键示例

```
CREATE TABLE deep_dive (
    aid         INT          --audience_id
   ,loc         CHAR(3)      --location
   ,dt          DATE         --date
) SORTKEY (dt, loc);
```

定义一个或多个列为排序键以对磁盘上的数据进行物理排序

deep_dive		
aid	loc	dt
1	SFO	2017-10-20
2	JFK	2017-10-20
3	SFO	2017-04-01
4	JFK	2017-05-14

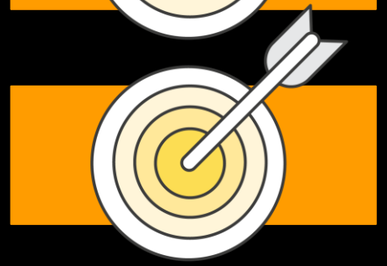
deep_dive (sorted)		
aid	loc	dt
3	SFO	2017-04-01
4	JFK	2017-05-14
2	JFK	2017-10-20
1	SFO	2017-10-20

Zone maps 以及排序示例

```
SELECT count(*) FROM deep_dive WHERE dt = '06-09-2017';
```

没有被排序的表

MIN: 01-JUNE-2017
MAX: 20-JUNE-2017



MIN: 08-JUNE-2017
MAX: 30-JUNE-2017

MIN: 12-JUNE-2017
MAX: 20-JUNE-2017

MIN: 02-JUNE-2017
MAX: 25-JUNE-2017

基于日期进行了排序

MIN: 01-JUNE-2017
MAX: 06-JUNE-2017

MIN: 07-JUNE-2017
MAX: 12-JUNE-2017

MIN: 13-JUNE-2017
MAX: 21-JUNE-2017

MIN: 21-JUNE-2017
MAX: 30-JUNE-2017

术语和概念 : Slices 切片

切片可以被认为是虚拟计算节点
- 数据分区单位
- 并行查询处理

关于切片的事实：
- 每个计算节点具有 2 或 16 个切片
- 表的数据分布在切片里
- 切片仅处理其自己的数据

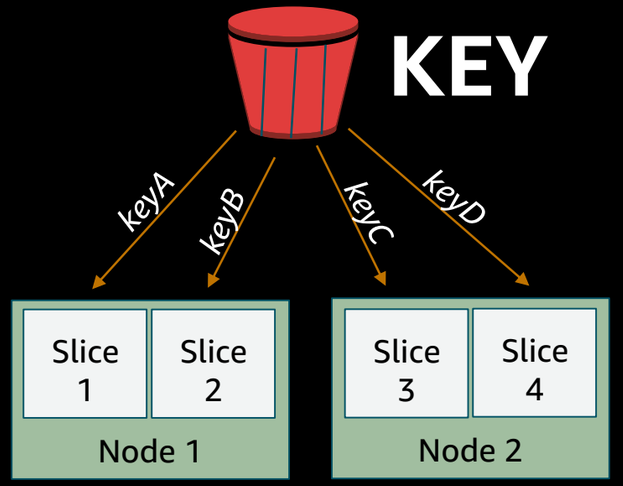
数据分配

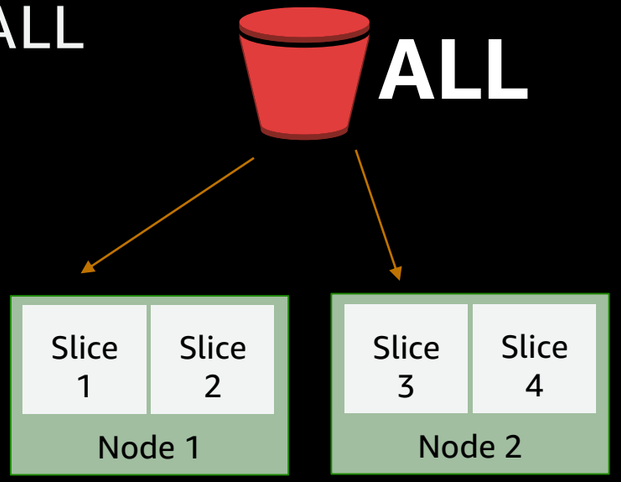
Distribution style 数据分配方式是一个表属性，它指示该表的数据如何在整个集群中分布：

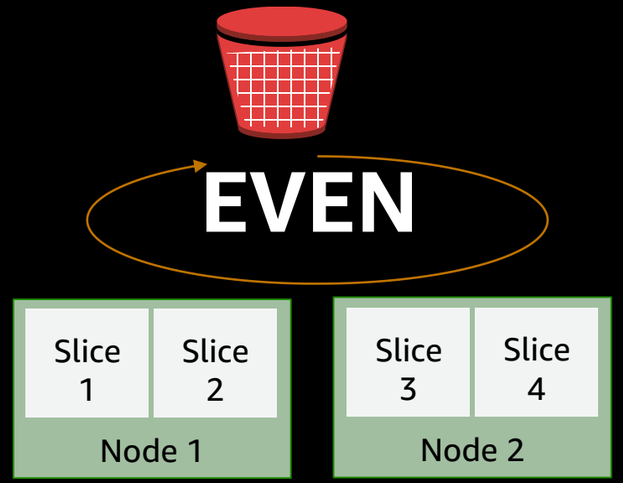
- KEY: 数值经过哈希计算后，相同哈希值的数据会落在想通的切片
- ALL: 全表数据冗余存储在每个计算节点的第一个切片上
- EVEN: 轮训方式（默认）
- AUTO: 自动结合 EVEN and ALL

目的:

- 更均匀的分配数据进行并行处理
- 查询过程中减少数据的移动



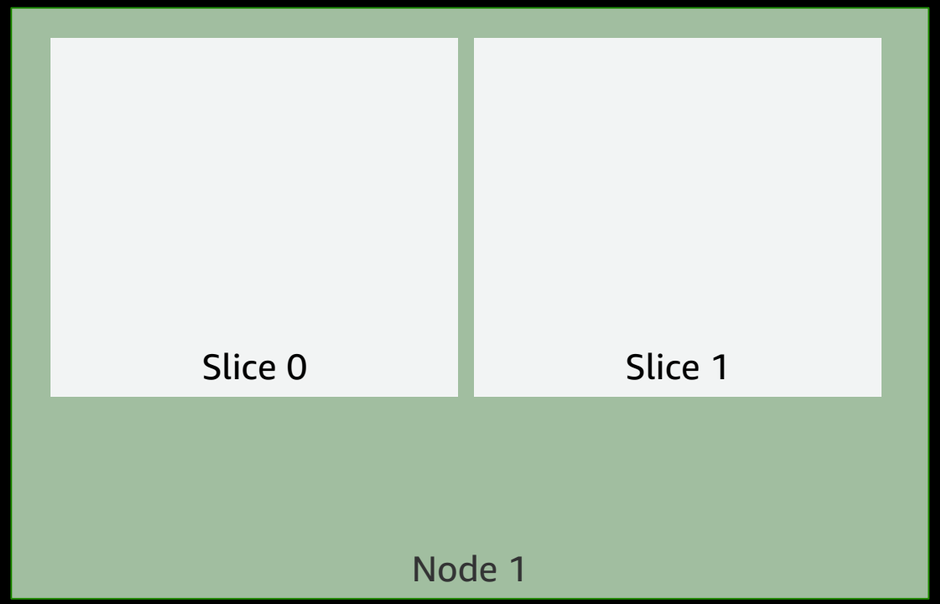


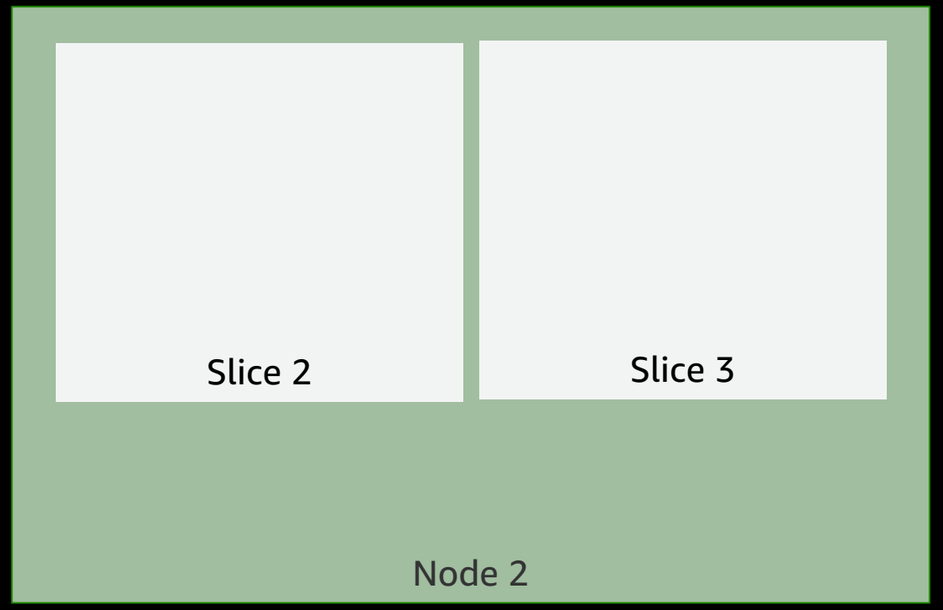


数据分配示例

```
CREATE TABLE deep_dive (
   aid        INT          --audience_id
   ,loc       CHAR(3)      --location
   ,dt        DATE         --date
) DISTSTYLE (EVEN|KEY|ALL|AUTO);
```

Table: deep_dive					
User Columns			**System Columns**		
aid	loc	dt	ins	del	row

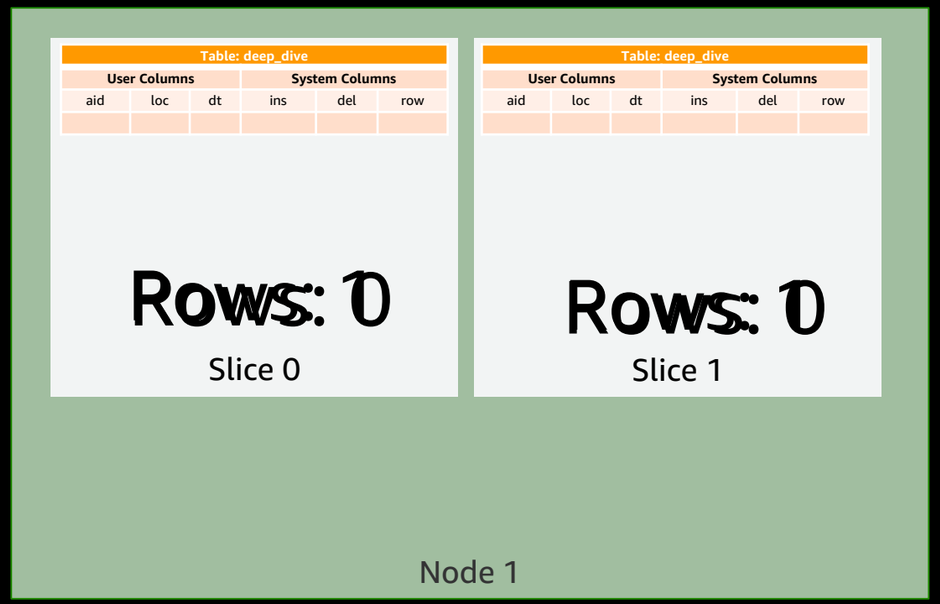


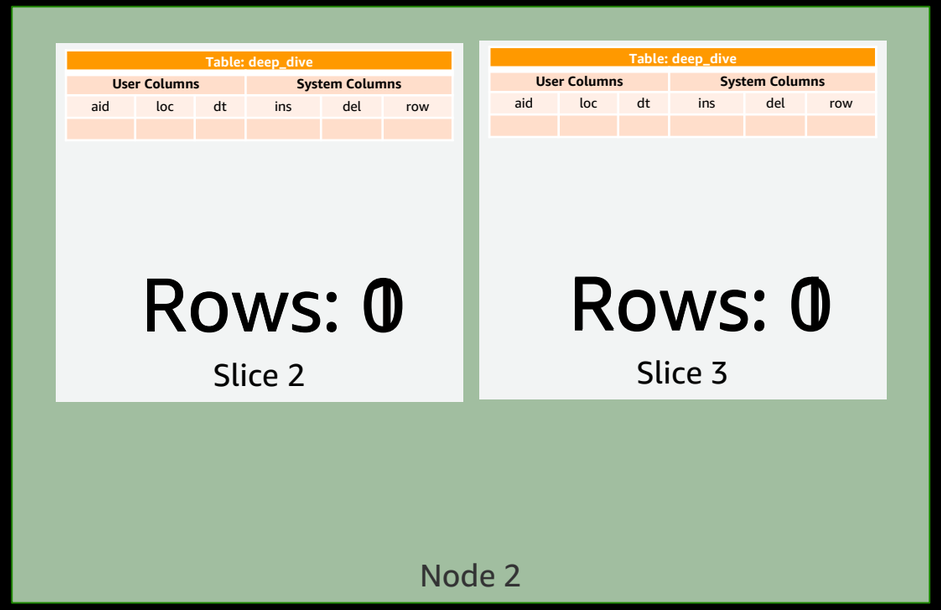


EVEN 数据分配示例

```
CREATE TABLE deep_dive (
   aid        INT          --audience_id
   ,loc       CHAR(3)      --location
   ,dt        DATE         --date
) DISTSTYLE EVEN;
```

```
INSERT INTO deep_dive VALUES
(1, 'SFO', '2016-09-01'),
(2, 'JFK', '2016-09-14'),
(3, 'SFO', '2017-04-01'),
(4, 'JFK', '2017-05-14');
```

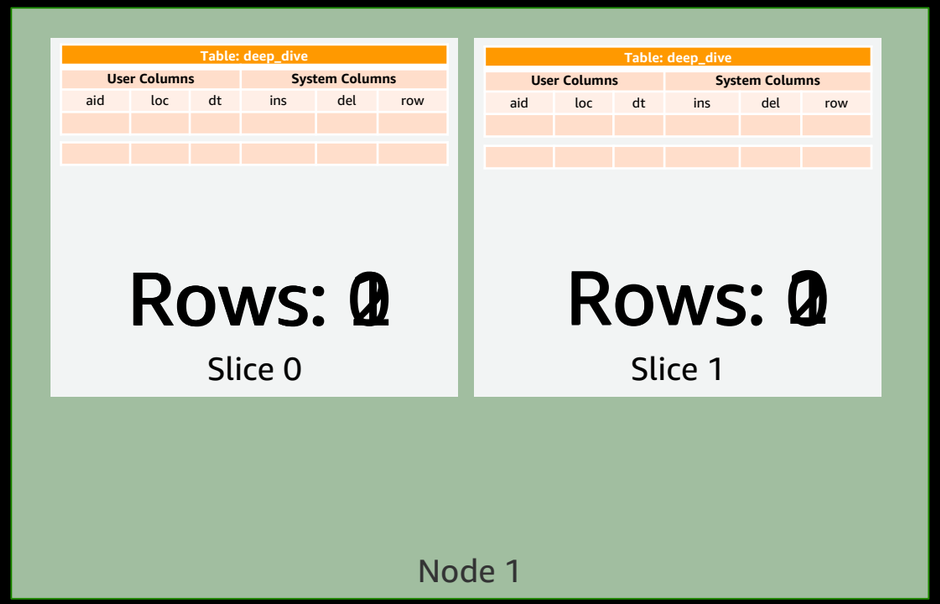


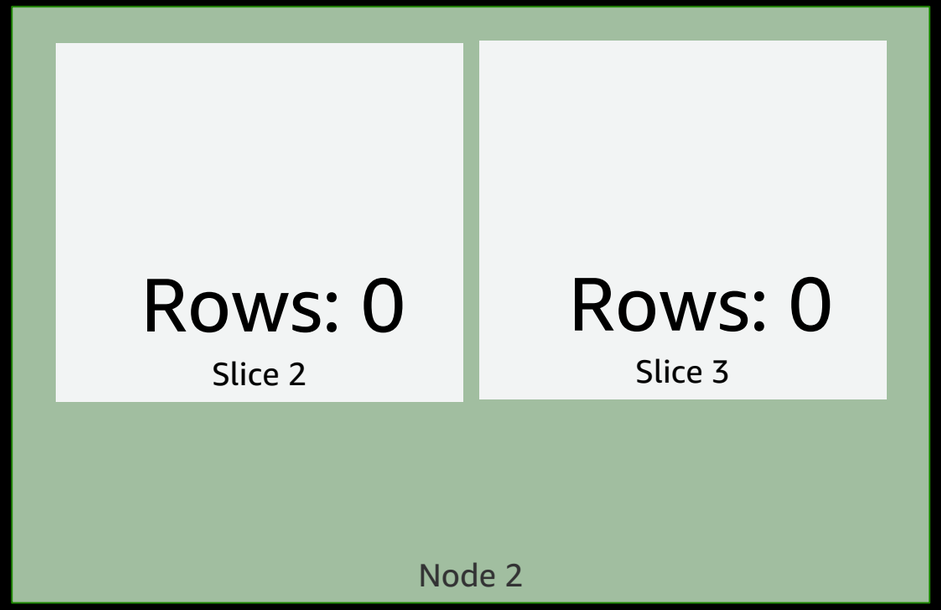


KEY 数据分配示例 #1

```
CREATE TABLE deep_dive (
   aid        INT          --audience_id
   ,loc       CHAR(3)      --location
   ,dt        DATE         --date
) DISTSTYLE KEY DISTKEY (loc);
```

```
INSERT INTO deep_dive VALUES
(1, 'SFO', '2016-09-01'),
(2, 'JFK', '2016-09-14'),
(3, 'SFO', '2017-04-01'),
(4, 'JFK', '2017-05-14');
```

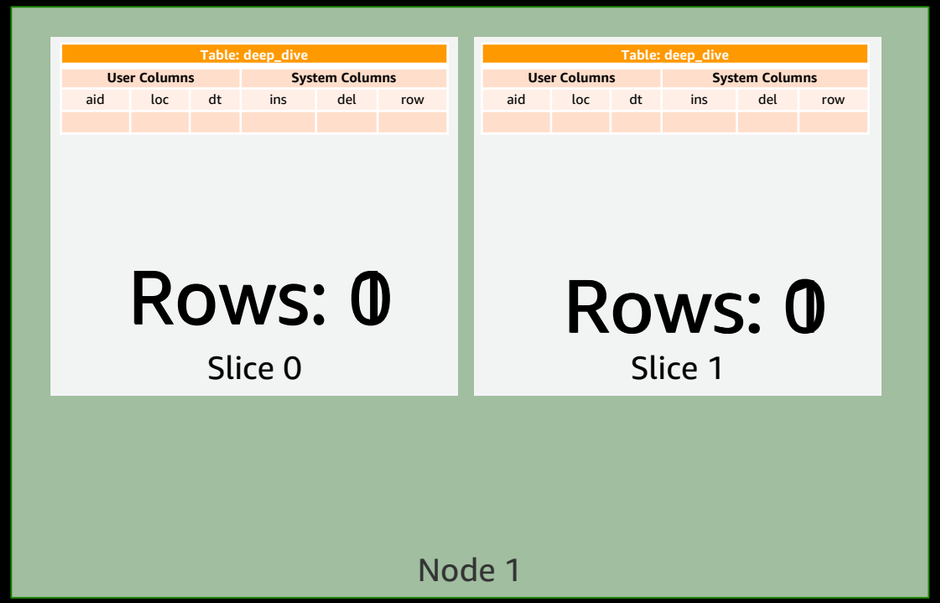


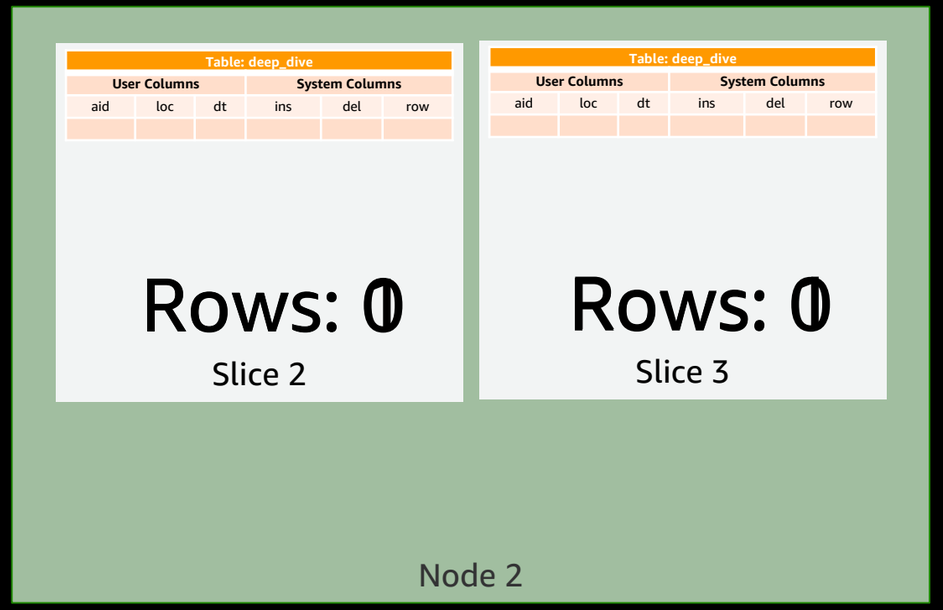


KEY 数据分配示例 #2

```
CREATE TABLE deep_dive (
   aid        INT          --audience_id
   ,loc       CHAR(3)      --location
   ,dt        DATE         --date
) DISTSTYLE KEY DISTKEY (aid);
```

```
INSERT INTO deep_dive VALUES
(1, 'SFO', '2016-09-01'),
(2, 'JFK', '2016-09-14'),
(3, 'SFO', '2017-04-01'),
(4, 'JFK', '2017-05-14');
```

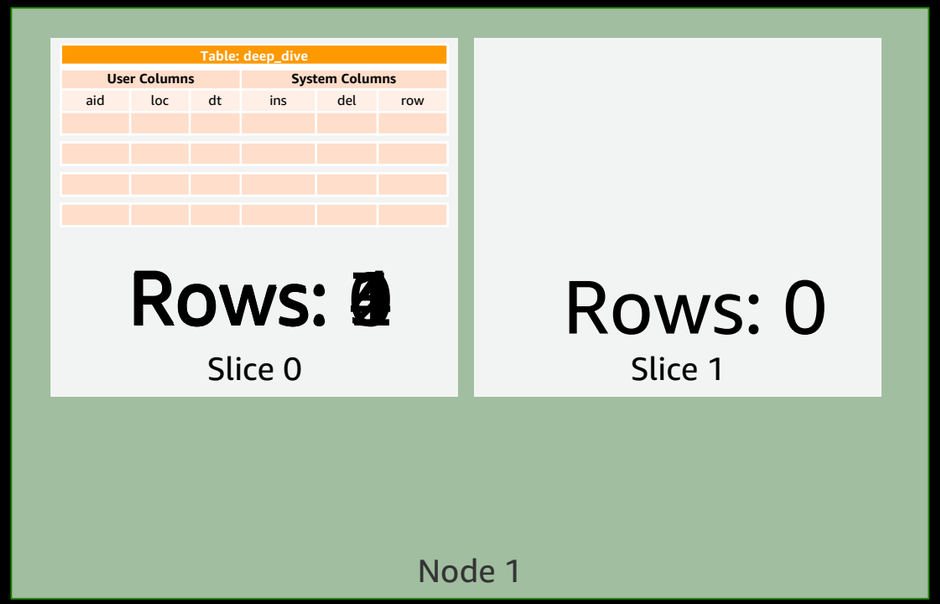


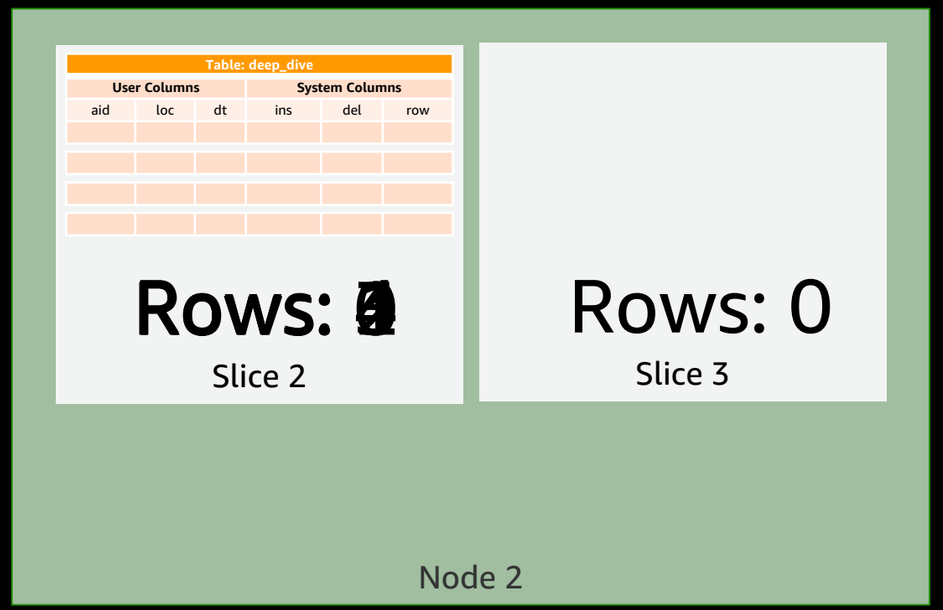


ALL 数据分配示例

```
CREATE TABLE deep_dive (
   aid        INT          --audience_id
   ,loc       CHAR(3)      --location
   ,dt        DATE         --date
) DISTSTYLE ALL;
```

```
INSERT INTO deep_dive VALUES
(1, 'SFO', '2016-09-01'),
(2, 'JFK', '2016-09-14'),
(3, 'SFO', '2017-04-01'),
(4, 'JFK', '2017-05-14');
```





数据分配总结

DISTSTYLE **KEY**

- 目的
  - 优化大表之间的 **JOIN** 操作性能，将 **ON** 两边的列作为各自表的分配 Key
  - 优化 **INSERT INTO SELECT** 性能
  - 优化 **GROUP BY** 性能
- 正在分布的列应该具有较高的基数，并且不会导致行倾斜。

DISTSTYLE **ALL**

- 目的
  - 优化和维度表之间的 **Join** 操作性能
  - 为小表操作减少磁盘 IO
  - 推荐针对中小维度表使用 (< 3M 行)

DISTSTYLE **EVEN**

- 如果 KEY 或 ALL 不适用

DISTSTYLE **AUTO**

- 新的默认分配方式- 结合 DISTSTYLE ALL and EVEN

表设计最佳实践总结

为列添加压缩

在筛选的主列上添加排序键

在不会造成数据倾斜的前提下使用 DISTSTYLE KEY 将大表数据分布在一起

避免对临时表使用分布

数据类型的长度按需规划，足够但不宜过长

- VARCHAR, CHAR, and NUMERIC

数据的存储和导入

aws INNOVATE
ONLINE CONFERENCE

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

术语和概念：数据冗余

Amazon Redshift 利用本地连接的存储设备

- 计算节点的存储容量是公布的 2.5 到 3 倍

全局 commit 确保所有永久表都已将块写入群集中的另一个节点，以确保数据冗余

异步备份块到 Amazon S3 始终一致的快照

- 每 5GB 的更改数据或每 8 小时
- 用户按需手动快照
- 在表级别禁用备份： **CREATE TABLE example(id int) BACKUP NO;**

Temporary tables 临时表

- 块未镜像到远程分区的速度比写性能快两倍
- 不触发完全 commit 或备份

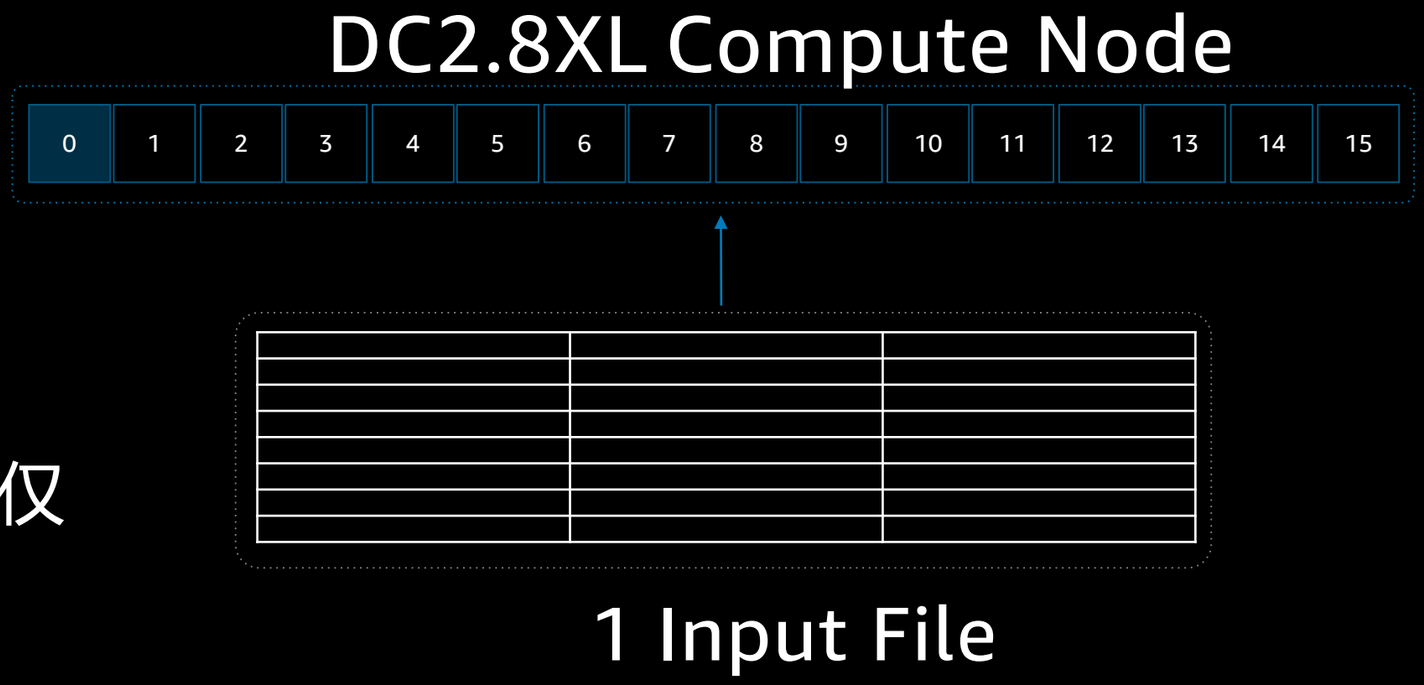
数据注入: COPY 命令

数据注入的吞吐量:

每个切片的查询处理器可以同时加载一个文件：

- 流式解压
- 格式解析
- 数据分配
- 数据写入

右边示例只实现部分节点使用，仅 6.25% 的切片处于活动状态



数据注入: COPY 命令

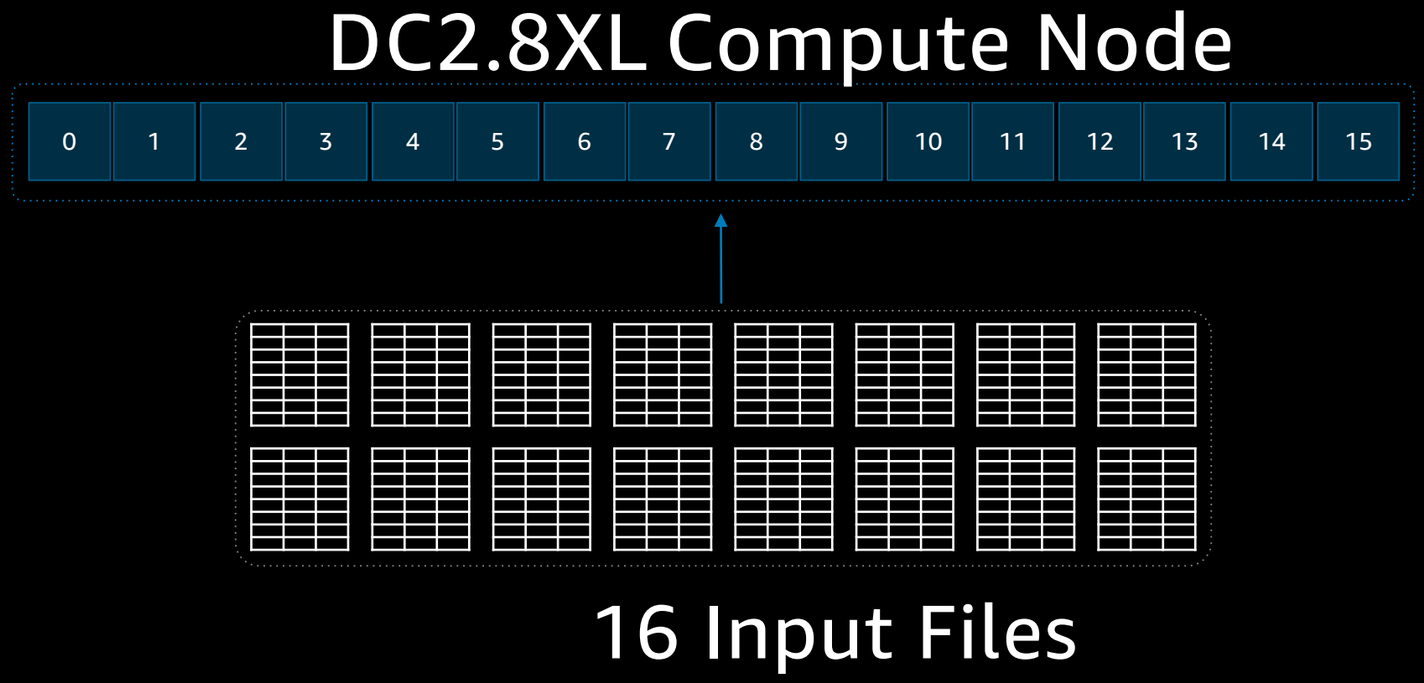
输入文件的数量应是切片数量的倍数

```
SELECT count(slice) from
stv_slices;
```

将单个文件拆分为 16 个输入文件，所有切片都在工作以最大限度地提高数据摄取性能

COPY 当添加节点时，复制性能将线性扩展

推荐使用 gzip 压缩后 —1 MB to 1 GB csv文件



数据注入: Amazon Redshift Spectrum

使用 INSERT INTO SELECT 从 Amazon S3 作为存储的外表选择数据
- 数据聚合
- 选择数据子集（列 / 行）
- 使用 SQL 添加新列

最佳实践：
- 将集群资源更多留给查询行负载而不是 ELT
- 与 COPY 相比 对数据的预先筛选和聚合能更有效提升效率

数据注入的设计考量

为大型写入设计

- 批量处理系统，优化处理海量数据
- 1 MB 大小加上不可变的数据块块意味着在写入时是按块克隆，以免引入碎片
- 小的写操作（约 1-10 行）与大的写操作（约 10 万行）的开销相似

UPDATE 和 DELETE

- 不可变的数据块意味着只是逻辑标注删除，在块尾写入新数据值
- 必须执行 VACUUM 或 DEEP COPY 才能从表里移除脏数据

VACUUM 和 ANALYZE

VACUUM 操作执行两类操作，将表全局排序以及清理被标记为删除的数据

ANALYZE 收集表统计信息以优化查询计划

最佳实践：

- VACUUM 应仅在必要时执行
  - 每周或夜晚
- ANALYZE 可以对 WHERE 操作经常筛选的列在数据注入后定期执行

Amazon Redshift 的崭新特性

aws INNOVATE
ONLINE CONFERENCE

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

不断演进的需求

数据的种类和数量
高速增长

集成
不同的数据集

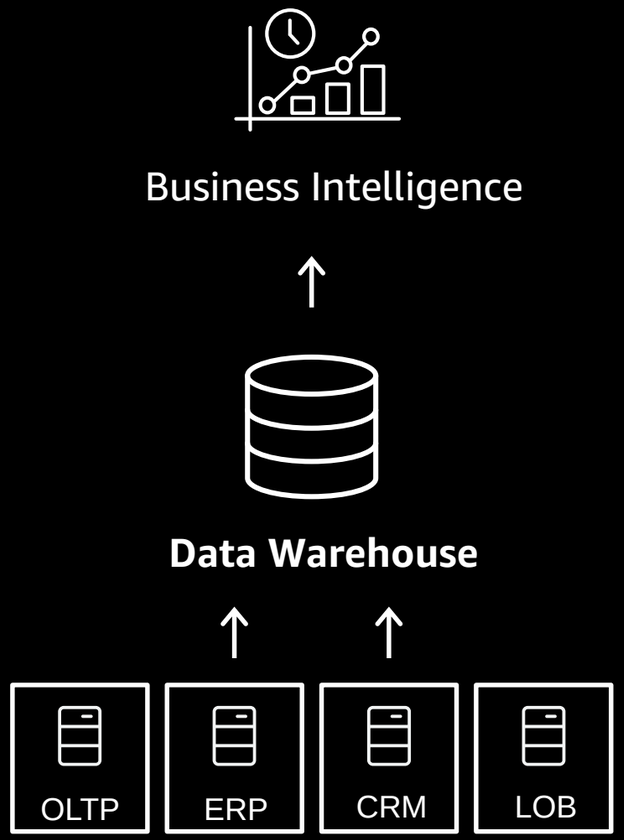
数据的自主访问
同时统一治理

数据分析从批处理
报表进化向
实时和预测

语音，图像识别，
物联网应用场景
被广泛纳入应用

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

曾经传统的数据分析体系



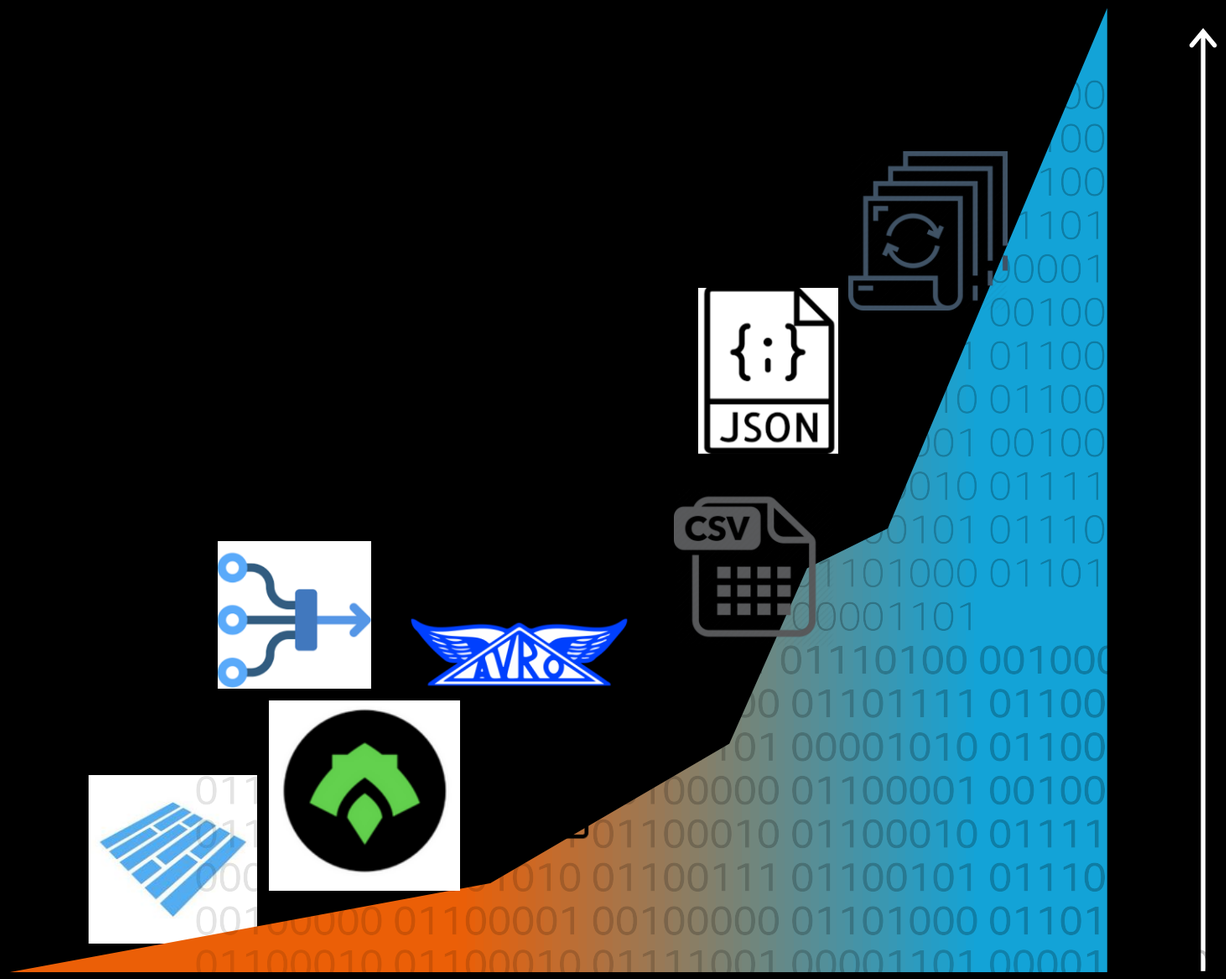
- 关系型数据
- GBs-TBs 数据规模
- 数据导入前定义数据结构
- 经营报表和即席 I 查询查询
- 非常大的初始资本支出

如今：比人们想象中更多的 更多的数据

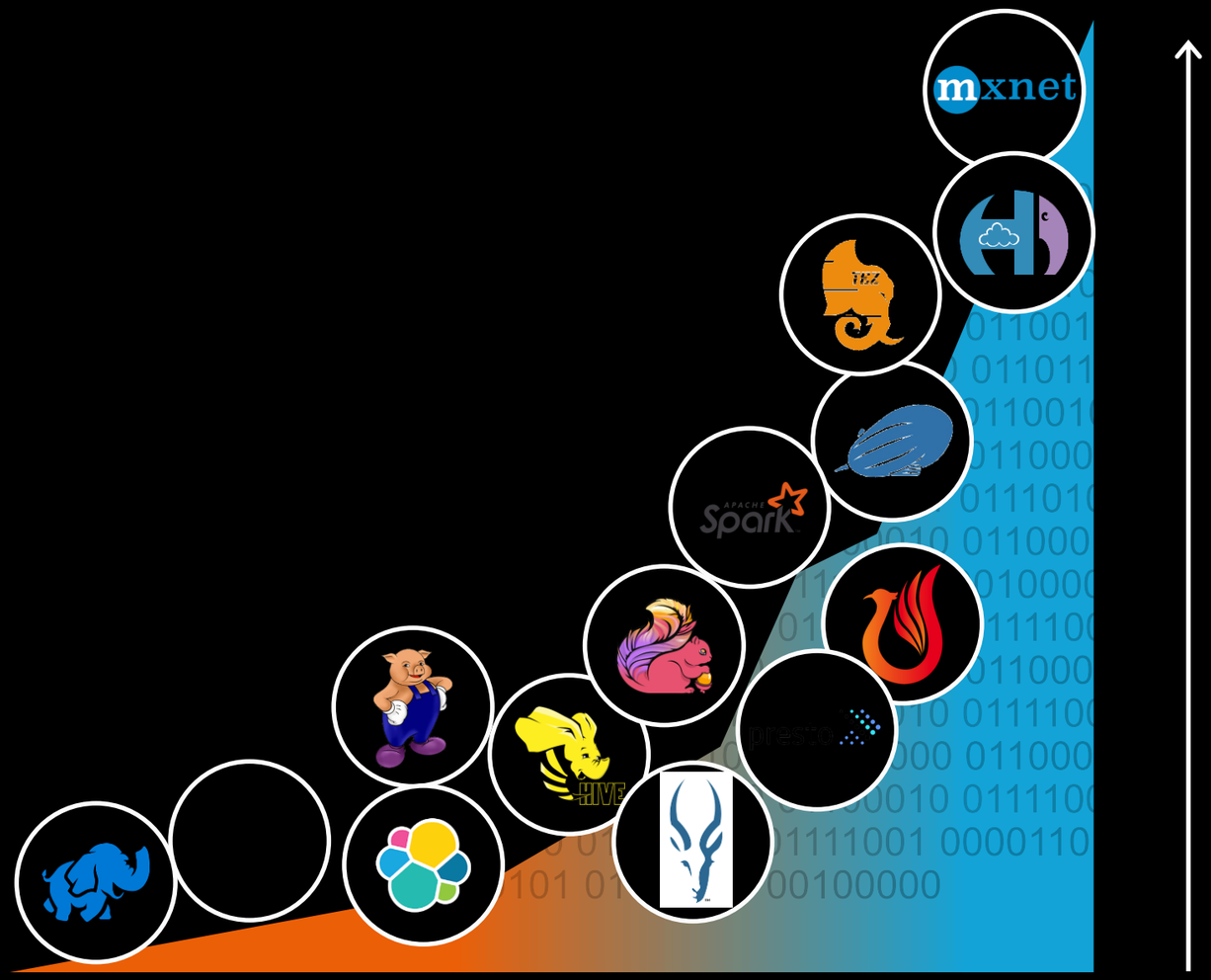
Data	Data platforms need to	
grows	live for	scale
>10x	15	1,000x
every 5 years	years	

* IDC, Data Age 20215: The Evolution of Data to Life-Critical Don't Focus on Big Data, Focus on the Data That's Big, April 2017.

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营



同时比过去**更多的数据类型**



市场上也有更多的手段和方法 去分析数据

11 — 8 — 5 — 4

Years ago

Hadoop	Elasticsearch	Presto	Spark
			Didn't exist

现代化数据仓库需要什么？

任何规模的数据，工作负载和用户

对不可预知的数据量和需求实现动态的扩展

更快得到见解

始终如一的快速性能，即使有数千个并发查询和用户

易于使用

不要在时间管理任务和维护上浪费时间

拓展你的数据湖

以开放格式直接分析存储在数据湖中的数据

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

Amazon Redshift

高效, 易用 经济 - 能将查询能力扩展到数据湖的数据仓库

分析开放数据格式 Parquet, ORC, and JSON, using SQL tools

高效

由机器学习，列式存储和 MPP 提供支持，为所有类型的分析工作负载提供更快的洞察时间

性能扩展

即使在不可预测的分析需求和数据量的情况下，也可以动态扩展以保证性能

数据湖的延伸

分析 Amazon S3 Data Lake 中的数据和开放格式，以及加载到 Redshift 高性能 SSD 中的数据

十分之一的成本

起价为每小时 0.25 美元，通过自动化管理任务节省成本，并消除因停机造成的业务影响; 每年每 TB 最低 1000 美元

Amazon Redshift

最新功能

Speed

- Improving short query acceleration
- NEW! Auto data distribution
- NEW! Elastic resize
- COMING SOON Concurrency Scaling

Scale

- COMING SOON Spectrum Request Accelerator
- COMING SOON Unload to Parquet

Simplicity

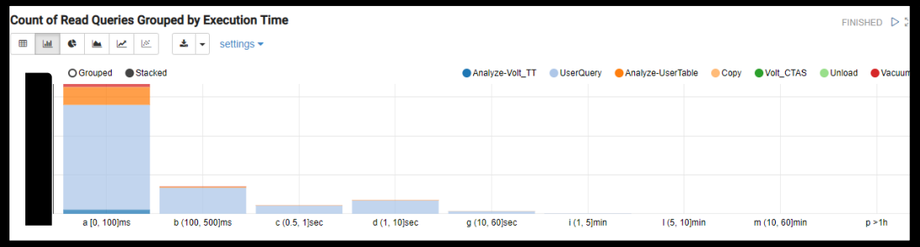
- COMING SOON WLM Concurrency Setting
- COMING SOON Auto-Vacuum & Auto-Analyze
- COMING SOON Support for stored procedures
- COMING SOON Auto Data Distribution
- NEW! Deferred Maintenance
- COMING SOON Snapshot Scheduler

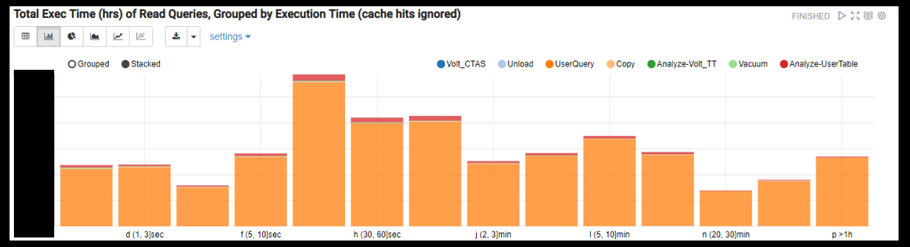
Security

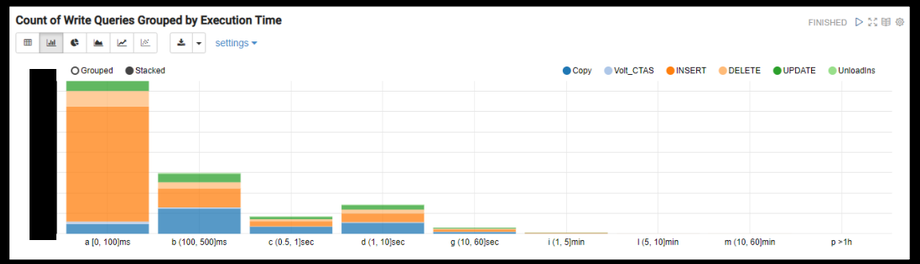
- COMING SOON Amazon Lake Formation integration

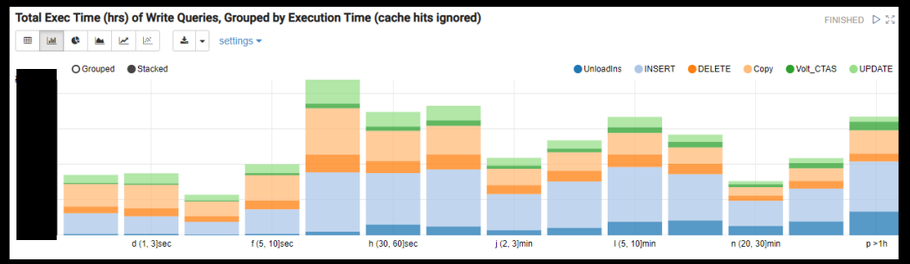
AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

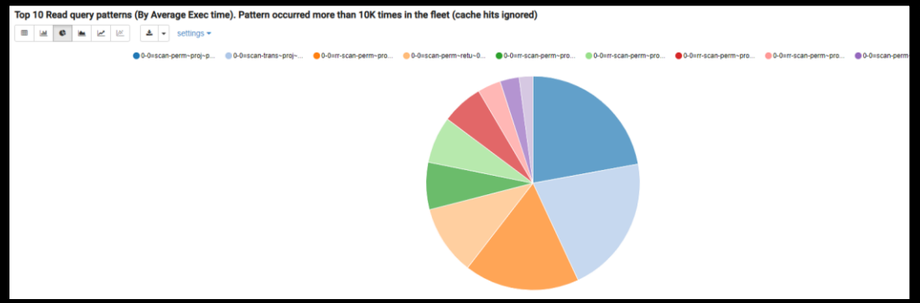
Amazon Redshift 使用指标监控

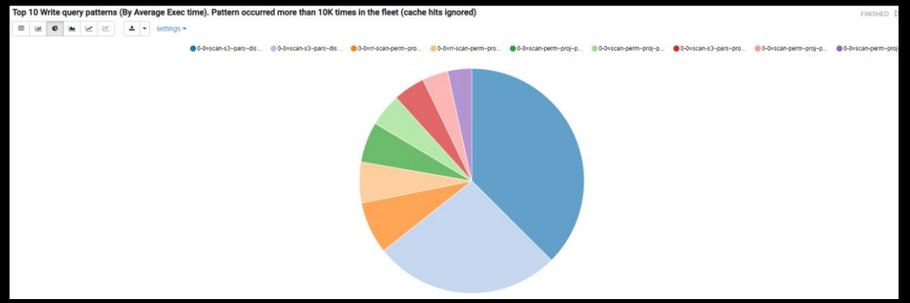








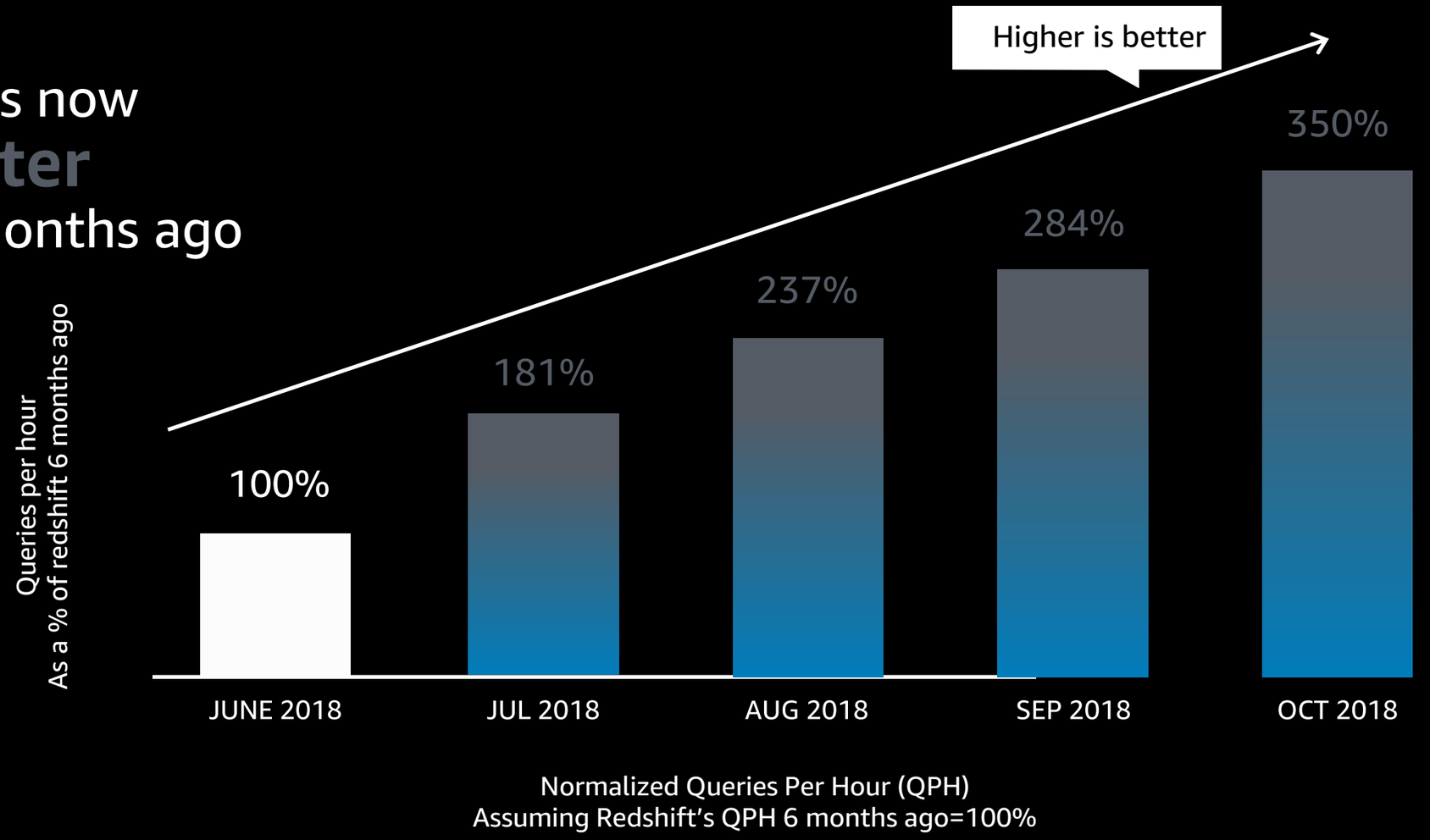




AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

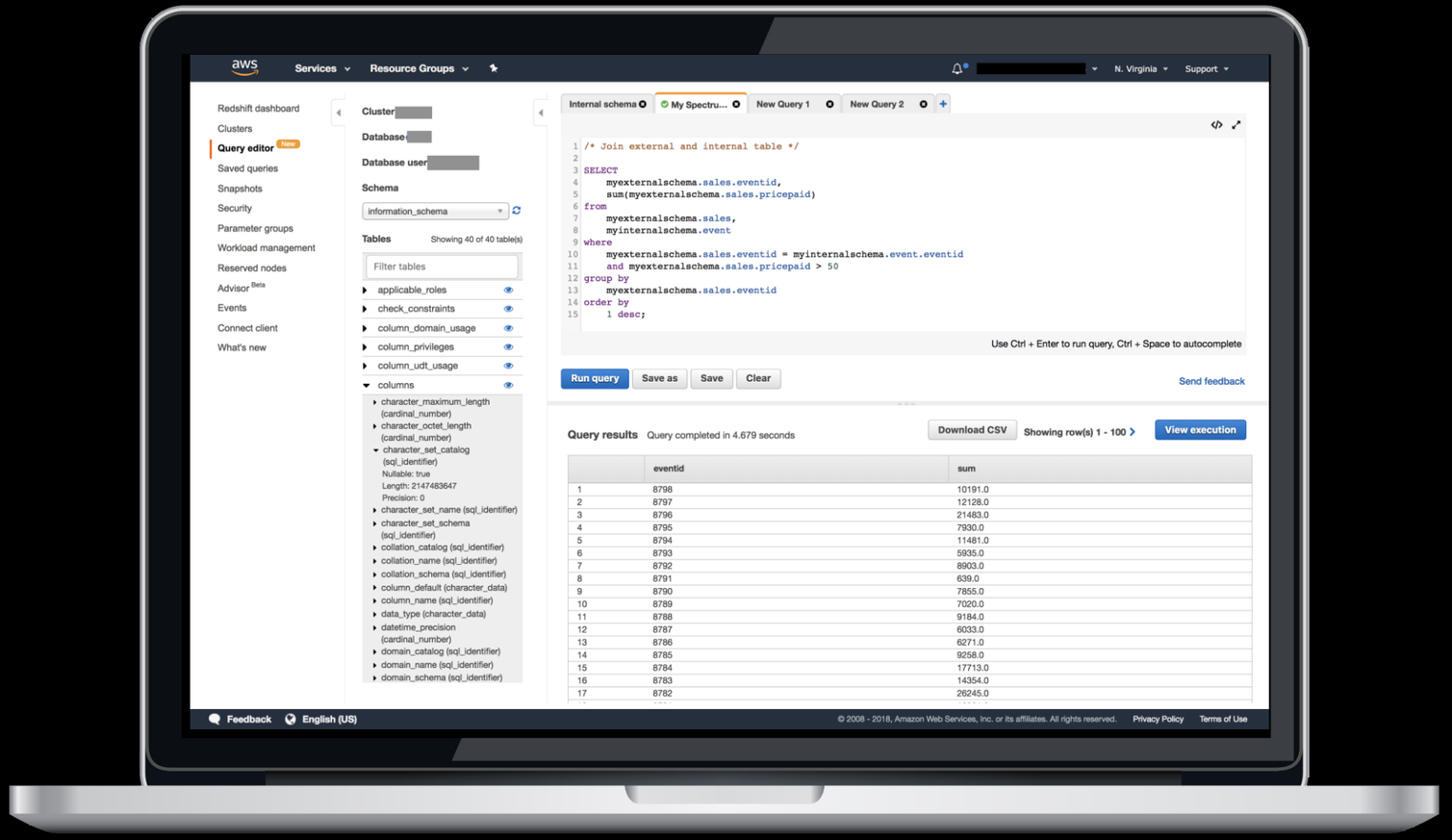
更快的洞察时间

Redshift is now
>3x faster
than 6 months ago



Normalized Queries Per Hour (QPH)
Assuming Redshift's QPH 6 months ago=100%

Redshift Query Editor



控制台直接执行查询

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

Redshift Advisor

提供自动优化推荐

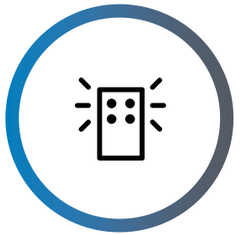
帮助优化数据库性能并降低运营成本

>96% 的集群

有定制的反馈

可操作的 WLM

COPY，存储， 和系统维护建议



智能建议

基于连续工作负载分析；优化性能

Amazon Redshift Spectrum

扩展数仓里的数据到 Amazon S3 数据湖 EB 级别的数据分析

无需导入

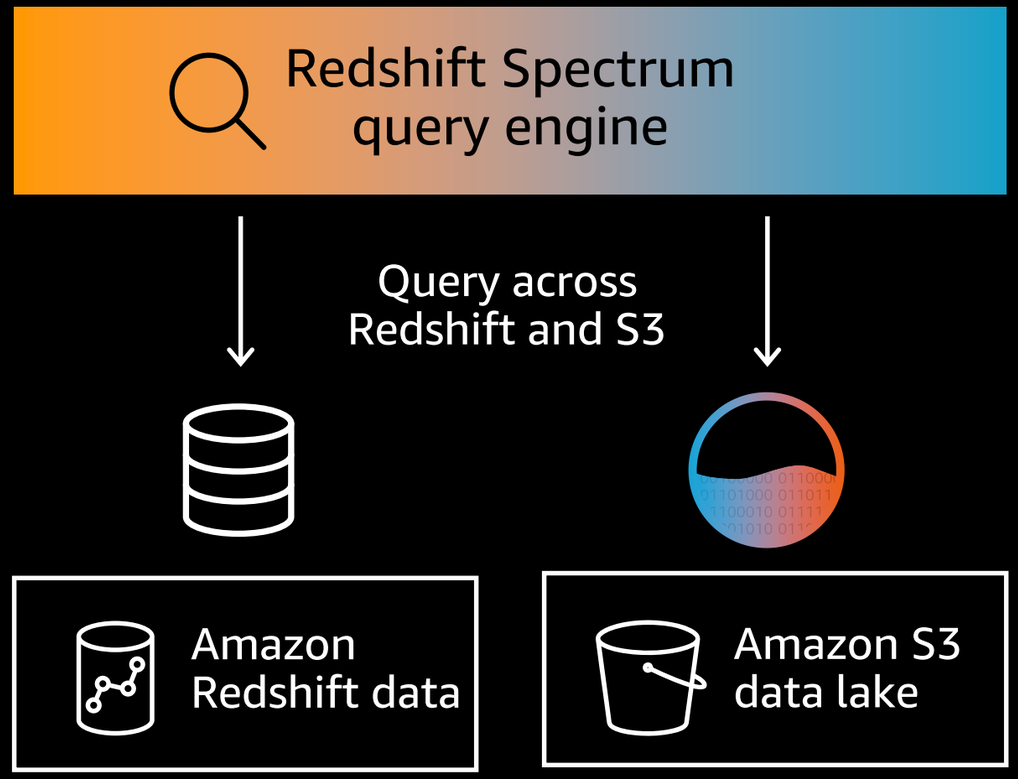
计算与存储分别扩展

直接查询 Amazon S3 数据

Parquet, ORC, Avro, Grok, and CSV 数据格式支持

→ 向 Amazon S3 Unload 成 Parque
→ Spectrum Request Accelerator

Coming Soon!



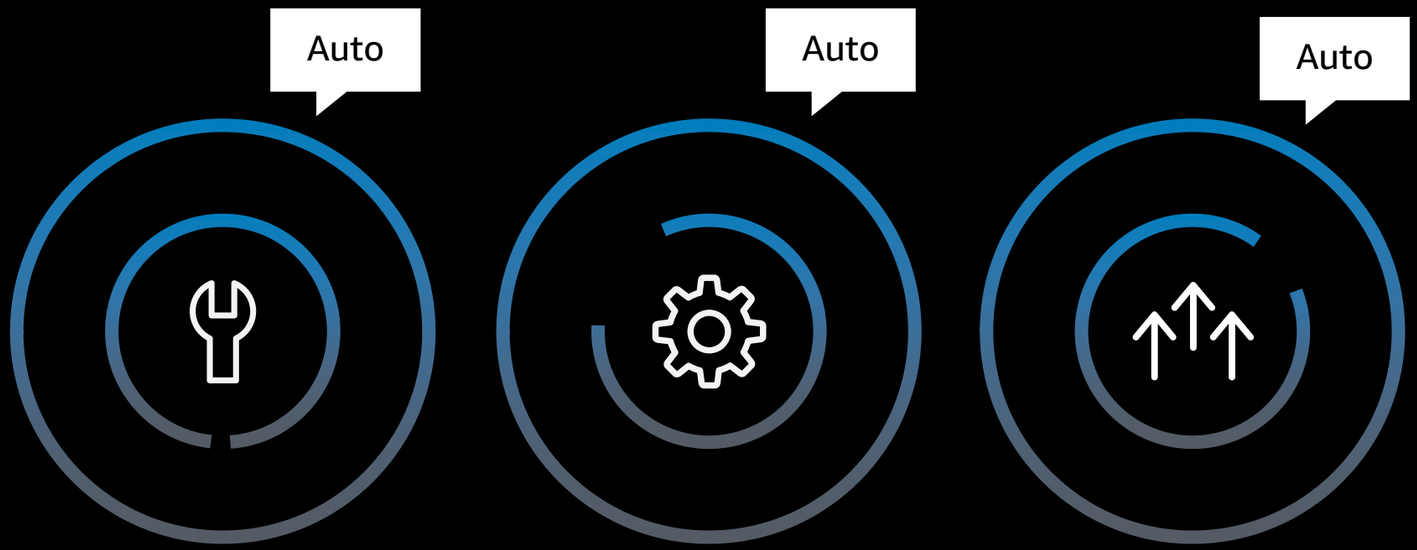
Amazon Redshift 智能维护

维护进程如 vacuum 和 analyze 会在后台自动执行

走向 zero-maintenance.

Amazon Redshift 自动调整 WLM 并行度设置以提供更好的吞吐.

Coming Soon!



Amazon Redshift Elastic Resize

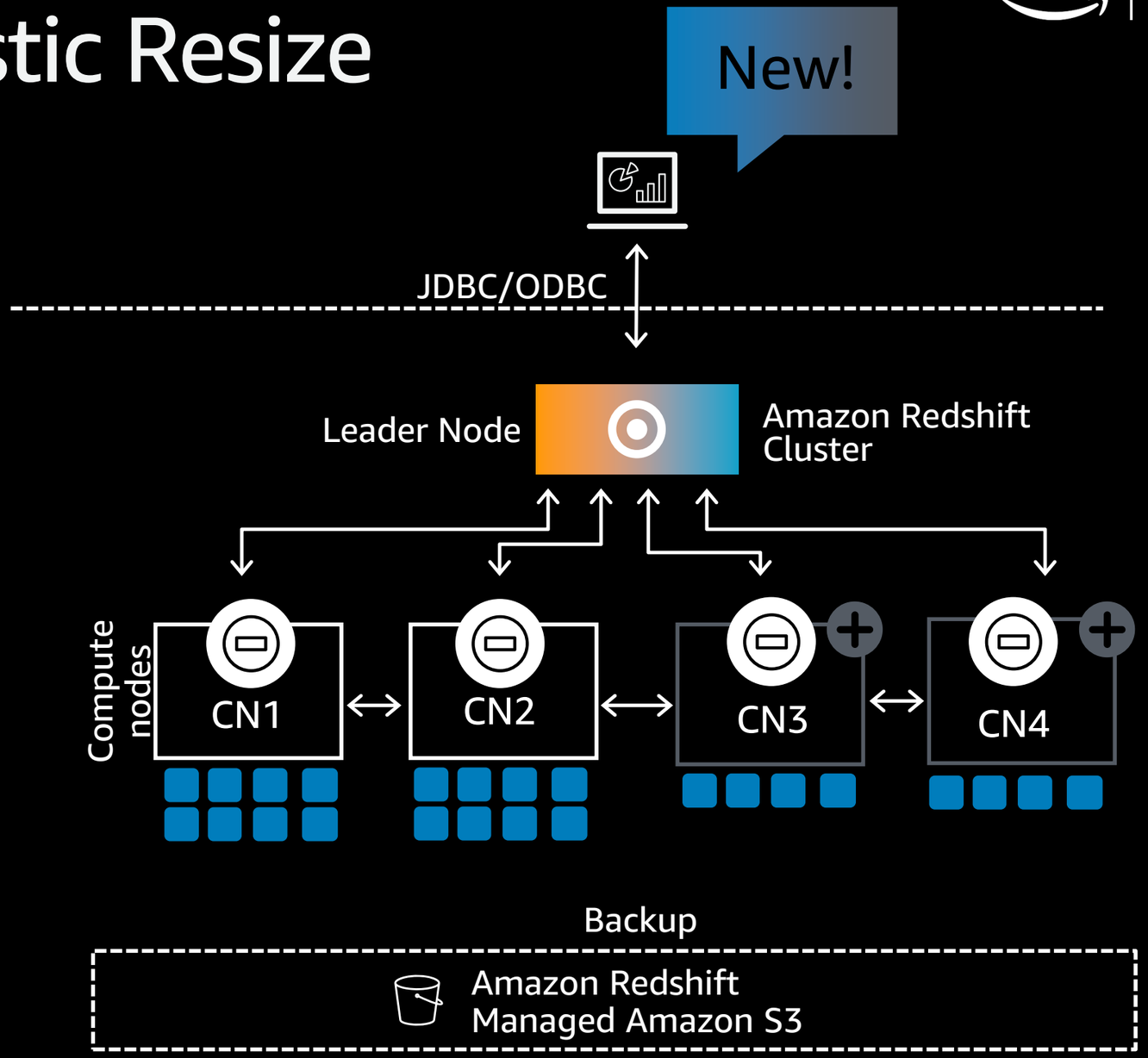
Scale up and down 分钟级扩展

添加新节点

更快
运行查询

最小化变换
过程时间

按需扩展计算及
存储资源



AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

Concurrency Scaling 应对用户活动高峰

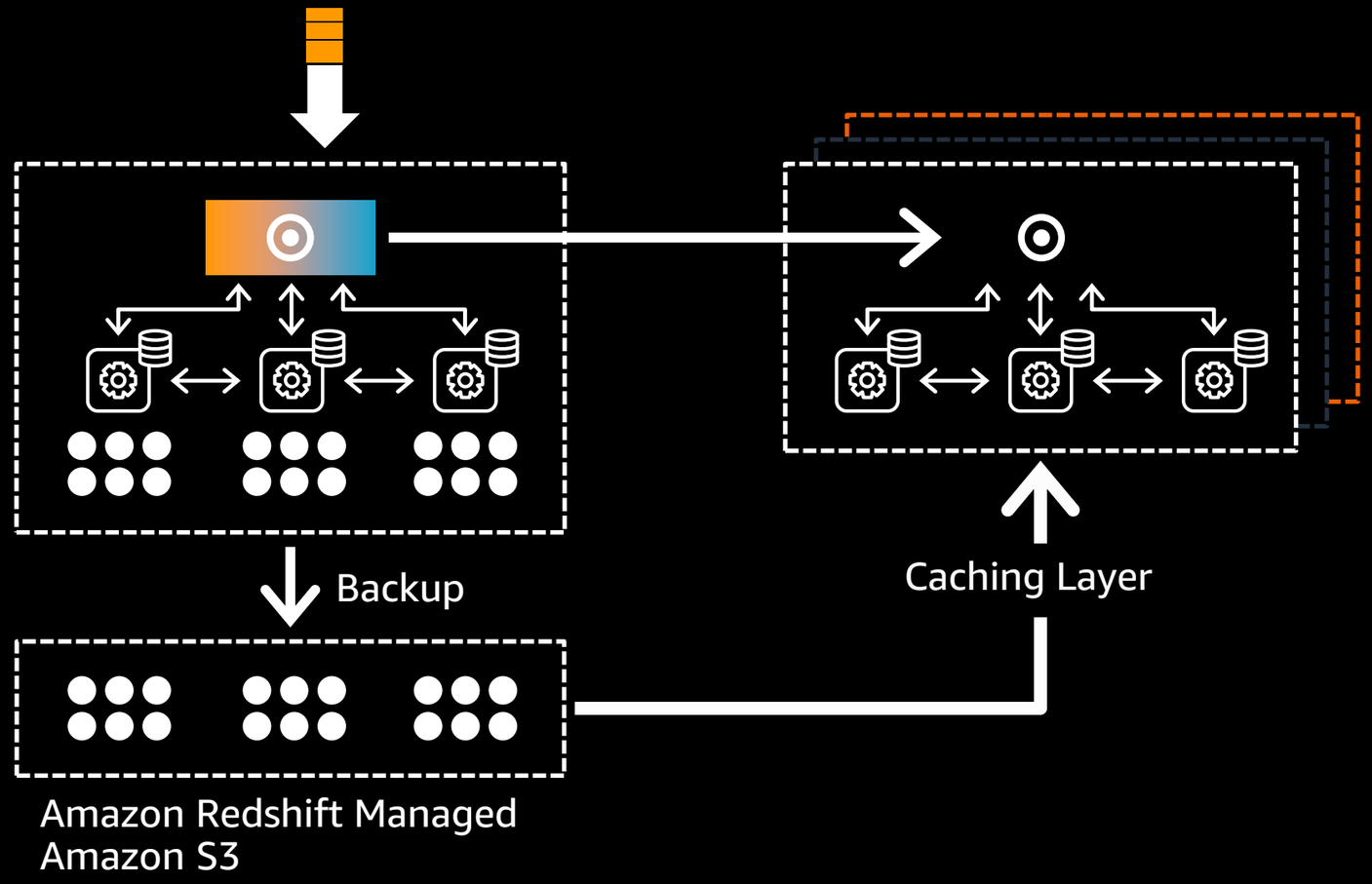
New!

自动创建 更多集群

一致稳定的性能
即使达到上千个并行查询

没有其它功能的牺牲让步

对 >97% 的客户免费
免费额度的积分制度，仅在高峰时期被消费



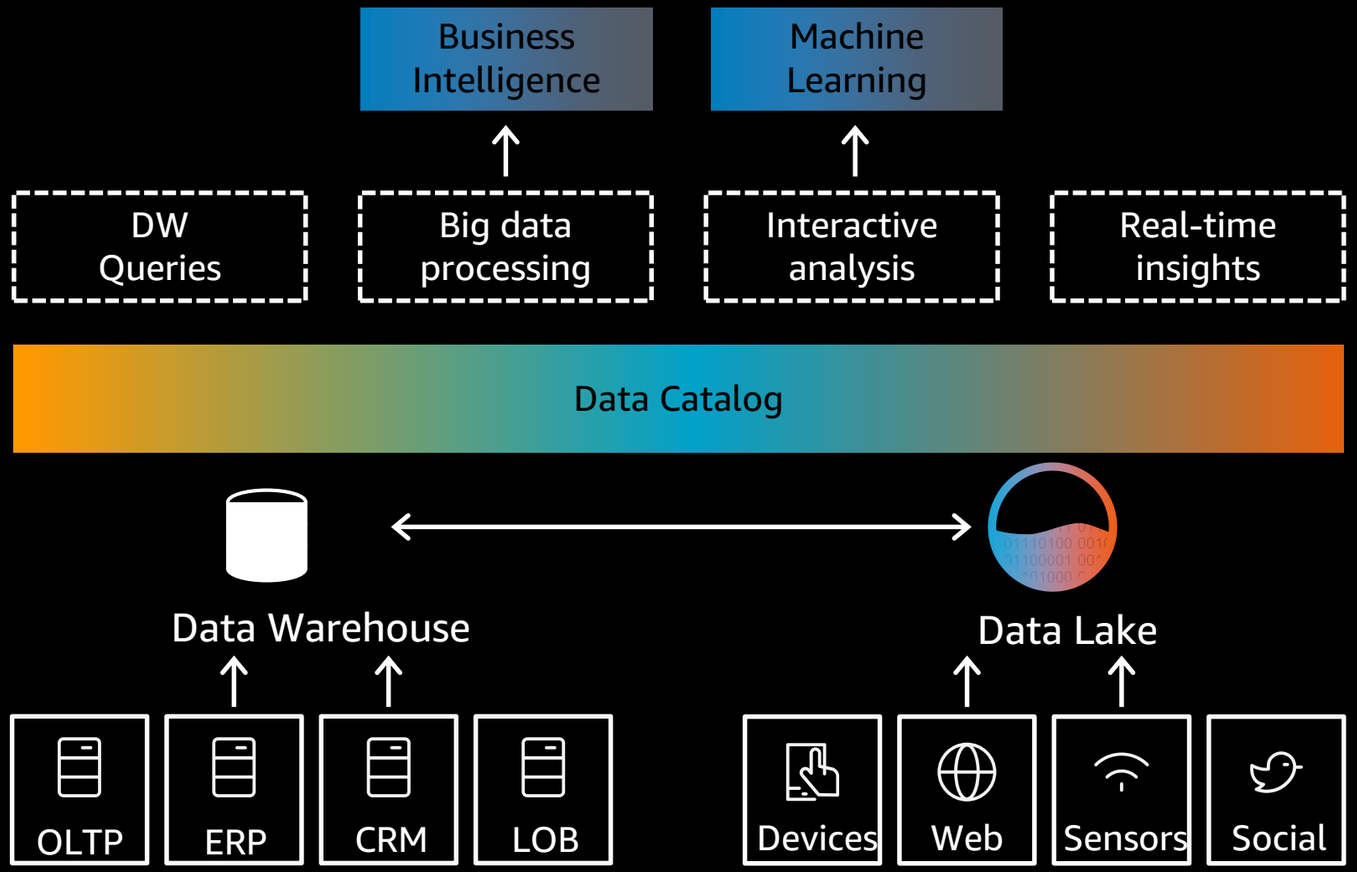
AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

构建现代化的数据分析链条

aws INNOVATE
ONLINE CONFERENCE

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

现代化的数据仓库同时意味着向数据湖架构转型

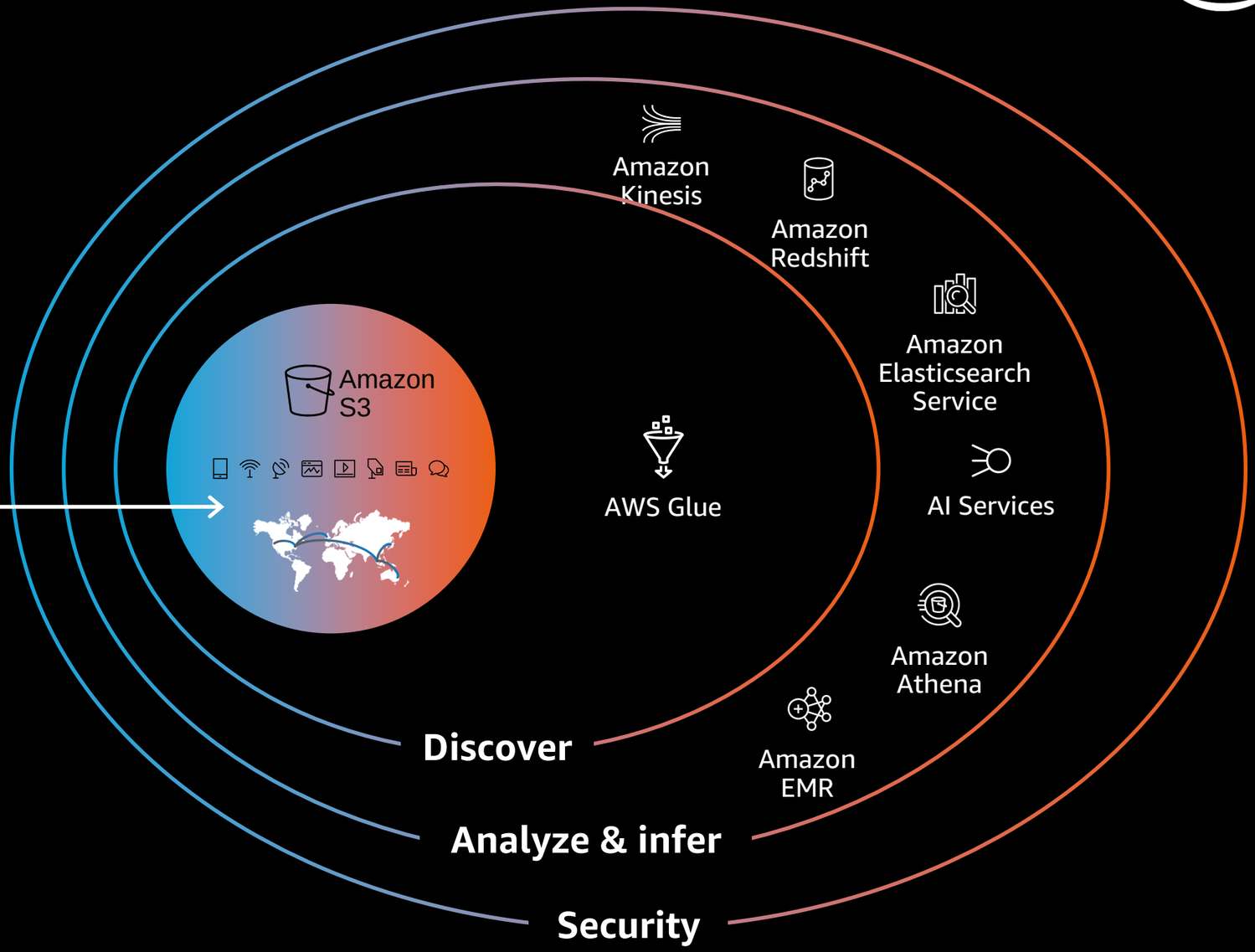


AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

数据湖多层结构

Ingest

- Amazon Kinesis Data Streams
- AWS Snowball
- Amazon Snowmobile
- Amazon Kinesis Data Streams
- Amazon Kinesis Data Firehose
- AWS Database Migration Service



AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

现代化您的数据仓库包括：

扩展查询能力到数据湖

与数据湖独立规划规模与容量

支持开放数据格式

与数据湖集成更多的分析工具

扩展性为先

统一的访问控制与数据治理

选择为未来十年+服务的解决方案

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

客户分享

re:Invent 2018 : Modern Cloud Data Warehousing ft. Equinox Fitness Clubs: Optimize Your Analytics Practices

EQUINOX

是一家以运动，营养和再生为中心的
综合奢侈品和生活方式产品的公司

blink FITNESS | PURE YOGA | SOULCYCLE | FURTHERMORE FROM EQUINOX | INTRODUCING EQUINOX HOTELS

Equinox 历史数据仓库

TERADATA

报表

“自助式”分析

用户画像

- 邮件营销
- CRM
- 定制化

数据科学不友好
成本
专业厂商知识

The “new” school

Doesn’t work for everything!

将所有数据存入 Hadoop 或 Amazon S3

不呀数据仓库

一切数据可以迟些用的时候再考虑如何取

数据仓库 vs 数据湖

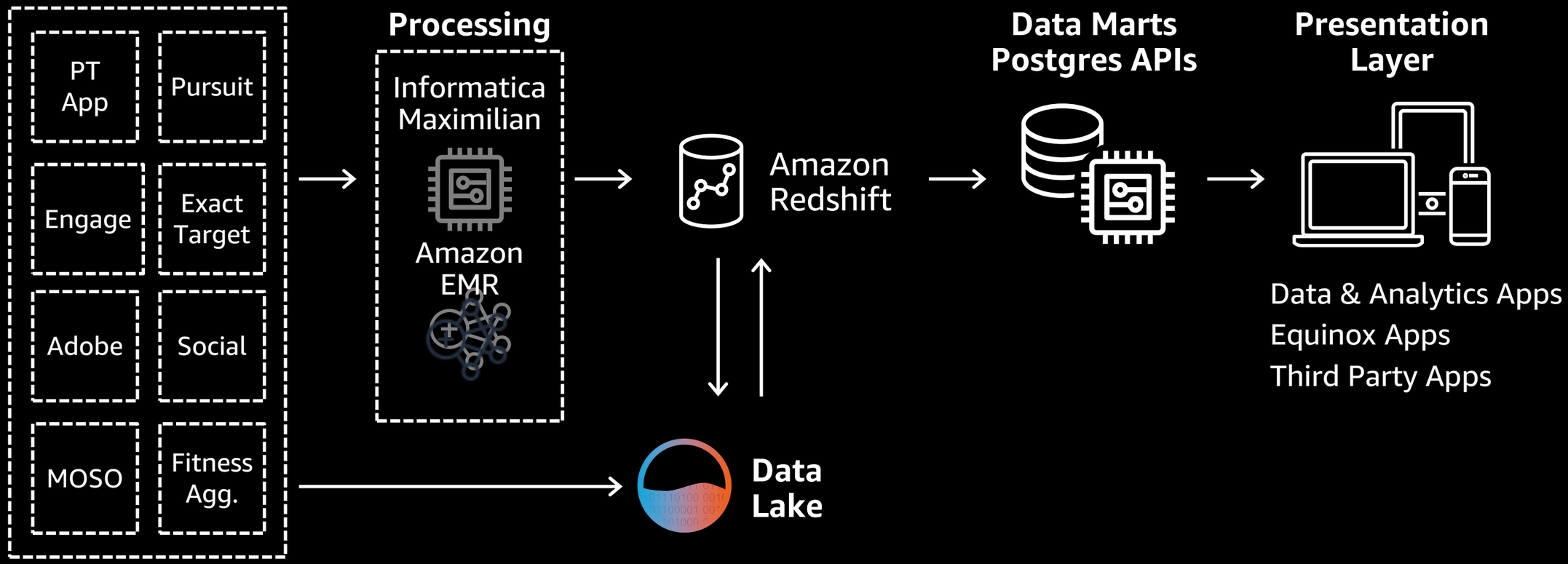
Data Lakes

海量不可变数据集
数据结构不敏感

Data Warehouse

可靠的高 SLA 报表系统
研发及分析师友好
特定的数据流程更加高效

JARVIS 项目架构



数据湖落地 pipeline 示例 ：Adobe Analytics

Query the data

Amazon Athena

Amazon Redshift

Amazon EMR

Metadata

AWS Glue

Storage

Amazon S3

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

数据湖落地 pipeline 示例 ：Adobe Analytics

Amazon Redshift 选型好处

大幅节省成本

学习成本

性能

DevOps 自动化

与 AWS 服务及第三方工具集成

https://www.youtube.com/watch?v=QZ4LAZCbsrQ

实用资源

aws INNOVATE
ONLINE CONFERENCE

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营

AWSLabs on GitHub — Amazon Redshift

https://github.com/awslabs/amazon-redshift-utils
https://github.com/awslabs/amazon-redshift-monitoring
https://github.com/awslabs/amazon-redshift-udfs

Admin scripts

Collection of utilities for running diagnostics on your cluster

Admin views

Collection of utilities for managing your cluster, generating schema DDL, and so on

Analyze Vacuum utility

Utility that can be scheduled to vacuum and analyze the tables within your Amazon Redshift cluster

Column Encoding utility

Utility that will apply optimal column encoding to an established schema with data already loaded

AWS Big Data Blog — Amazon Redshift

Amazon Redshift Engineering's Advanced Table Design Playbook

https://aws.amazon.com/blogs/big-data/amazon-redshift-engineerings-advanced-table-design-playbook-preamble-prerequisites-and-prioritization/

- Zach Christopherson

Top 10 Performance Tuning Techniques for Amazon Redshift

https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-techniques-for-amazon-redshift/

- Ian Meyers and Zach Christopherson

Twelve Best Practices for Amazon Redshift Spectrum

https://aws.amazon.com/blogs/big-data/10-best-practices-for-amazon-redshift-spectrum/

- Po Hong and Peter Dalton

感谢参加 AWS INNOVATE 2019 在线技术大会

我们希望您在这里找到感兴趣的内容！

也请帮助我们完成**投票打分**和**反馈问卷**。

欲获取关于 AWS 的更多信息和技术内容，可以通过以下方式找到我们：


微信公众号：AWSChina


新浪微博：https://www.weibo.com/amazonaws/


领英：https://www.linkedin.com/company/aws-china/


知乎：https://www.zhihu.com/org/aws-54/activities/


视频中心：http://aws.amazon.bokecc.com/


更多线上活动：https://aws.amazon.com/cn/about-aws/events/webinar/

aws INNOVATE
ONLINE CONFERENCE

AWS 中国（宁夏）区域由西云数据运营
AWS 中国（北京）区域由光环新网运营