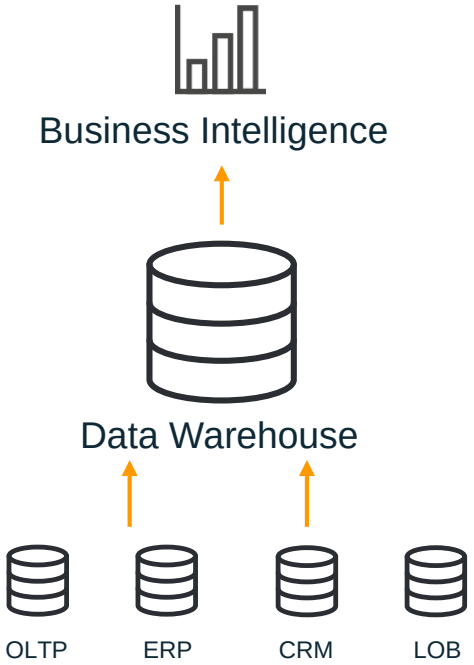
Data Warehousing and Data Lake Analytics, Together

Ben Snively, Solutions Architect – Data and Analytics

May 23rd, 2018

aws

Data Warehouse...



Relational data

Gigabytes to Petabytes scale

Reporting and analysis

Schema defined prior to data load

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Amazon Redshift

a lot faster
a lot easier to use
a lot cheaper

Relational data warehouse

Massively parallel; Petabyte scale

Fully managed

HDD and SSD Platforms

$1,000/TB/Year; starts at $0.25/hour

aws

Nasdaq

Nasdaq operates financial exchanges around the world, and processes large volumes of data.

Challenge:

Nasdaq wanted to make their large historical data footprint available to analyze as a single dataset.

Solution:

- Use Amazon Redshift for interactive querying
- 5.5 billion rows into Amazon Redshift every day (peaking at 14 billion one day)

Philips Uses AWS to Power a 37-Million Record Upload to the Cloud in 90 Minutes

> Thanks to AWS, our business was able to upload over 37 million records into Amazon Redshift in 90 minutes.
>
> **Doug Ranaham**
> Manager, Philips Healthcare

- Philips' U.S.-based healthcare division needed to analyze 37 million records for a new project
- When its on-premises database solution couldn't handle a data transfer that large anymore, the company turned to AWS and the AWS Marketplace
- Philips set up Attunity CloudBeam for Amazon Redshift in less than one minute to upload 37 million records to the AWS Cloud in 90 minutes
- Using AWS, Philips can optimize any size data set within two hours, allowing its consultants and analysts to provide solutions quickly to customers

aws

Amazon Redshift

10x faster at 1/10th the cost

Easy to Use

Create and start using a data warehouse in minutes

Fast

Delivers fast results for all types of workloads

Cost-effective

No upfront costs, start small, and pay as you go

Scalable

Gigabytes to petabytes to exabytes

Integrated

Integrated with S3 data lakes, AWS services and third-party tools

Secure

Audit everything; encrypt data end-to-end; extensive certification and compliance

aws

Amazon Redshift is easy to use

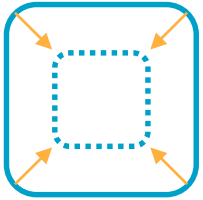
Provisioning in minutes

Automatic patching

SQL - Data loading

Backups are built-in

Security is built-in



Compression is built-in

aws

Amazon Redshift is fast

REDFIN

*"Did I mention that it's **ridiculously fast**? We're using it to provide our analysts with an alternative to Hadoop"*

Periscope DATA

*"After investigating Redshift, Snowflake, and BigQuery, we found that Redshift offers **top-of-the-line performance at best-in-market price points**"*

boingo

*"On our previous big data warehouse system, it took around 45 minutes to run a query against a year of data, but that number went down to **just 25 seconds** using Amazon Redshift"*

Pinterest

*"…[Amazon Redshift] performance has blown away everyone here. We generally see **50-100X speedup over Hive**"*

CHANNEL FOUR TELEVISION

*"We **regularly process multibillion row datasets and we do that in a matter of hours**. We are heading to up to 10 times more data volumes in the next couple of years, easily"*

optimum.

*"We saw a **2X performance improvement** on a wide variety of workloads. The **more complex the queries, the higher the performance improvement**"*

aws

Amazon Redshift is integrated

Use Existing BI Tools

Use your existing cloud/on-premises Business Intelligence (BI) Tools

Amazon Redshift

JDBC/ODBC

Tool based native drivers

Existing or new BI tool

VPC

aws

On Prem

Periscope Data

Amazon QuickSight

tableau

OpenText Analytics

looker

Qlik

SISENSE

Dundas BI

JReport

ClearStory Data

birst

Logi Analytics

CHARTIO

IBM Business Partner

DOMO

pentaho A Hitachi Group Company

TIBCO

MODE

GoodData

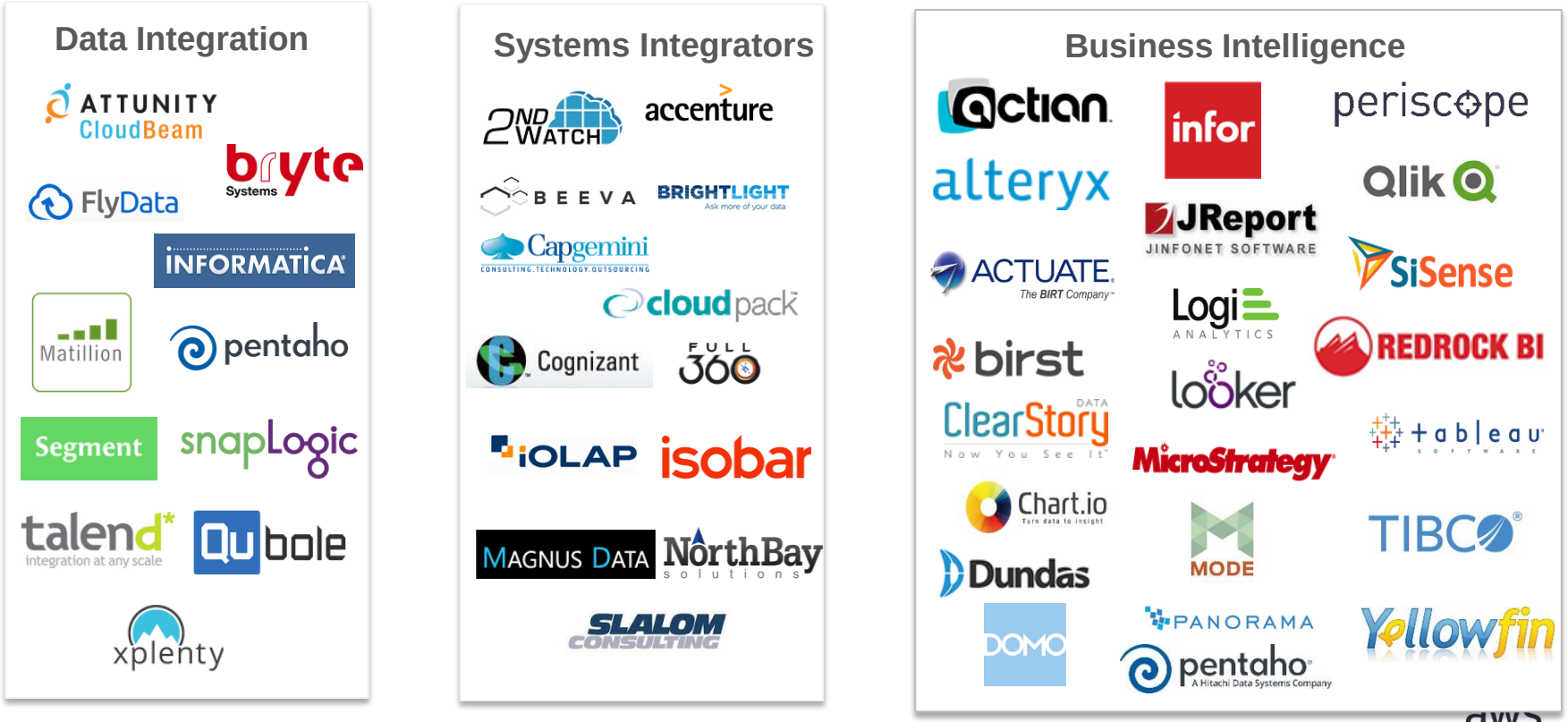
MicroStrategy

Yellowfin

PANORAMA

aws

Amazon Redshift is integrated



Accelerate migrations from legacy systems

"AWS Database Migration Service is the most impressive migration service we've seen." Gartner

Migrate – Over 1,000 unique migrations to Amazon Redshift using AWS DMS

aws

Redshift is used for mission-critical workloads

Financial and management reporting

Payments to suppliers and billing workflows

Web/Mobile clickstream and event analysis

Recommendation and predictive analytics

aws

Diving into Redshift

Columnar Architecture: Example

```
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;
```

Column-based storage

- Only scan blocks for relevant column

aws

Compression: Example

```
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

More efficient compression is due to storing the same data type in the columnar architecture

Columns grow and shrink independently

Reduces storage requirements

Reduces I/O

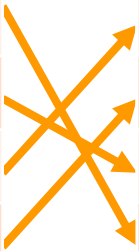
aws

Sort Key: Example

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

Add a sort key to one or more columns to physically sort the data on disk

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



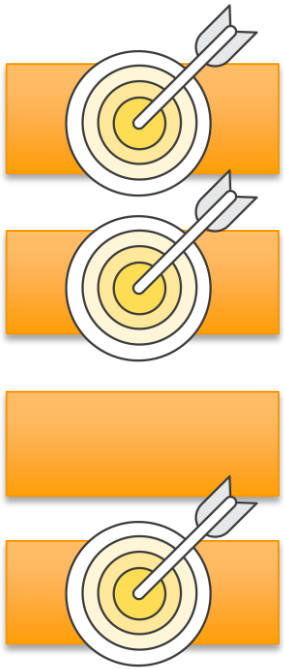
deep_dive (sorted)		
aid	loc	dt
3	SFO	2017-04-01
4	JFK	2017-05-14
2	JFK	2017-10-20
1	SFO	2017-10-20

aws

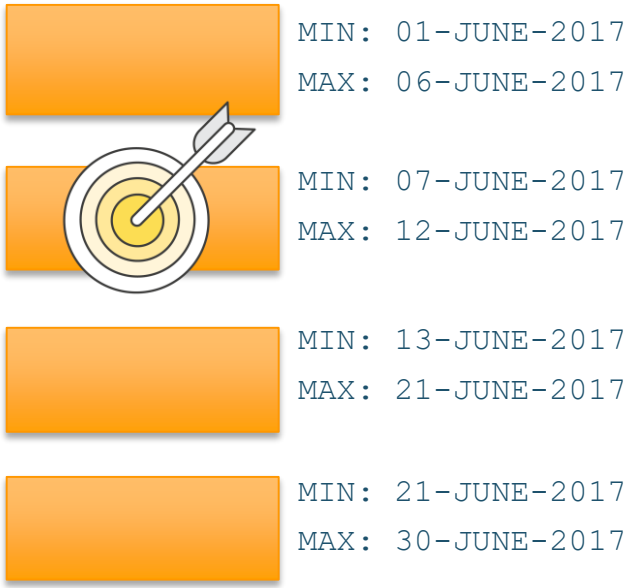
Zone Maps and Sorting: Example

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

Unsorted table



Sorted by date



aws

Terminology and Concepts: Slices

A *slice* can be thought of like a virtual compute node

- Unit of data partitioning
- Parallel query processing

Facts about slices:

- Each compute node has either 2, 16, or 32 slices
- Table rows are distributed to slices
- A slice processes only its own data

aws

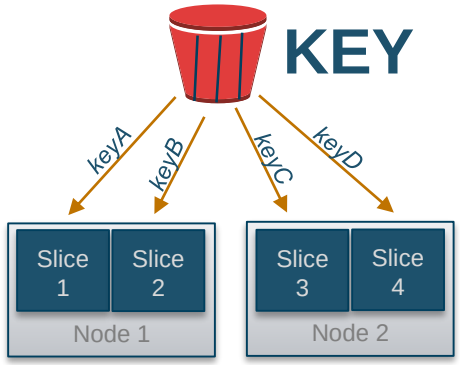
Data Distribution

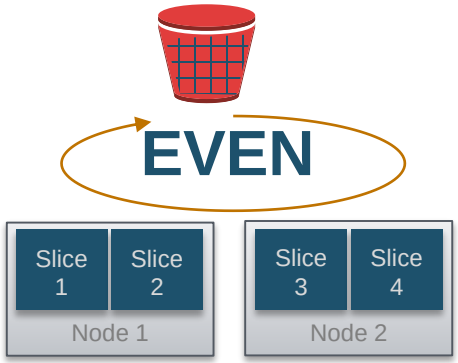
Distribution style is a table property which dictates how that table's data is distributed throughout the cluster:

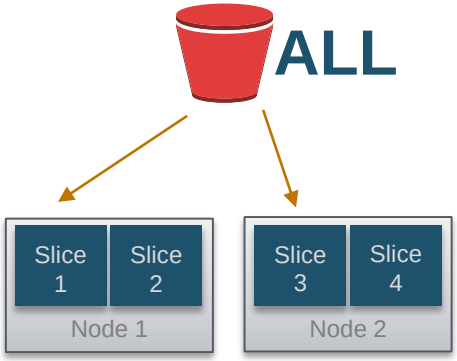
- KEY: value is hashed, same value goes to same location (slice)
- ALL: full table data goes to the first slice of every node
- EVEN: round robin

Goals:

- Distribute data evenly for parallel processing
- Minimize data movement during query processing







aws

Best Practices: Data Distribution

DISTSTYLE **KEY**

- Goals
  - Optimize **Join** performance between large tables
  - Optimize **Insert into Select** performance
  - Optimize **Group By** performance
- The column that is being distributed on should have a high cardinality and not cause row skew:

```
SELECT diststyle, skew_rows FROM svv_table_info WHERE "table" = 'deep_dive';
 diststyle | skew_rows
-----------+-----------
 KEY(aid)  |      1.07
```

← Ratio between the slice with the most and least number of rows

DISTSTYLE **ALL**

- Goals
  - Optimize **Join** performance with dimension tables
  - Reduces disk usage on small tables
  - Small and medium size dimension tables (< 3M rows)

DISTSTYLE **EVEN**

- If neither KEY or ALL apply (or you are unsure)

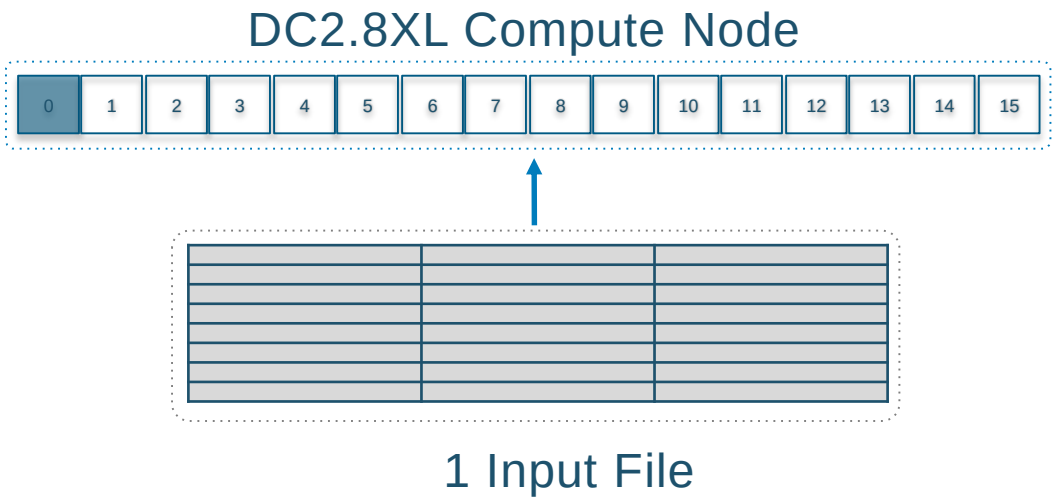
aws

Data Ingestion: COPY Statement

Ingestion throughput:

- Each slice's query processors can load one file at a time:
  - Streaming decompression
  - Parse
  - Distribute
  - Write

Realizing only partial node usage as 6.25% of slices are active



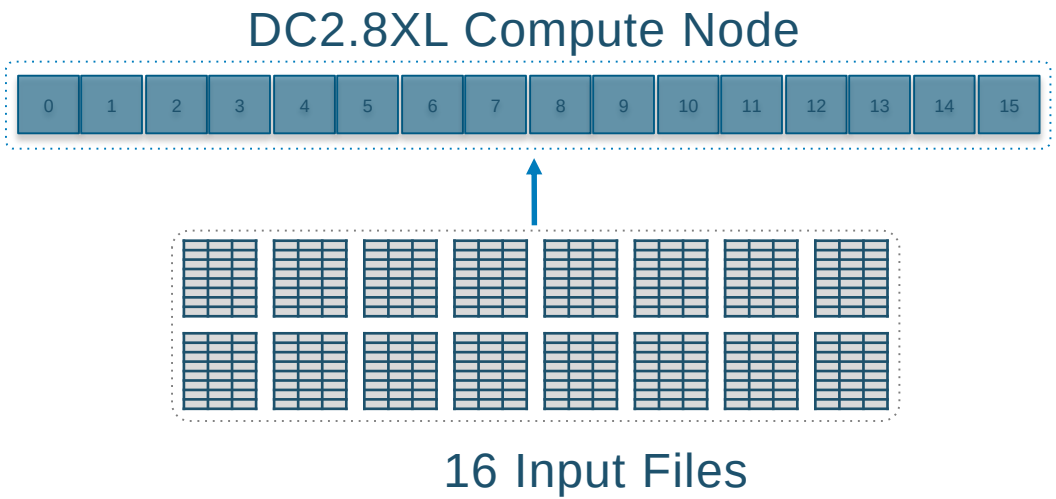
aws

Data Ingestion: COPY Statement

Number of input files should be a multiple of the number of slices

Splitting the single file into 16 input files, all slices are working to maximize ingestion performance

COPY continues to scale linearly as you add nodes



Recommendation is to use delimited files—1 MB to 1 GB after gzip compression

aws

Node Types and Cluster Sizing

Terminology and Concepts: Node Types

Dense Compute—DC2

- Solid state disks

Dense Storage—DS2

- Magnetic disks

Instance Type	Disk Type	Size	Memory	CPUs
DC2 large	NVMe SSD	160 GB	16 GB	2
DC2 8xlarge	NVMe SSD	2.56 TB	244 GB	32
DS2 xlarge	Magnetic	2 TB	32 GB	4
DS2 8xlarge	Magnetic	16 TB	244 GB	36

aws

Best Practices: Cluster Sizing

Use at least two computes nodes (multi-node cluster) in production for data mirroring

- Leader node is given for no additional cost

Amazon Redshift is significantly faster in a VPC compared to EC2 Classic

Maintain at least 20% free space or three times the size of the largest table

- Scratch space for usage, rewriting tables
- Free space is required for vacuum to re-sort table
- Temporary tables used for intermediate query results

The maximum number of available Amazon Redshift Spectrum nodes is a function of the number of slices in the Amazon Redshift cluster

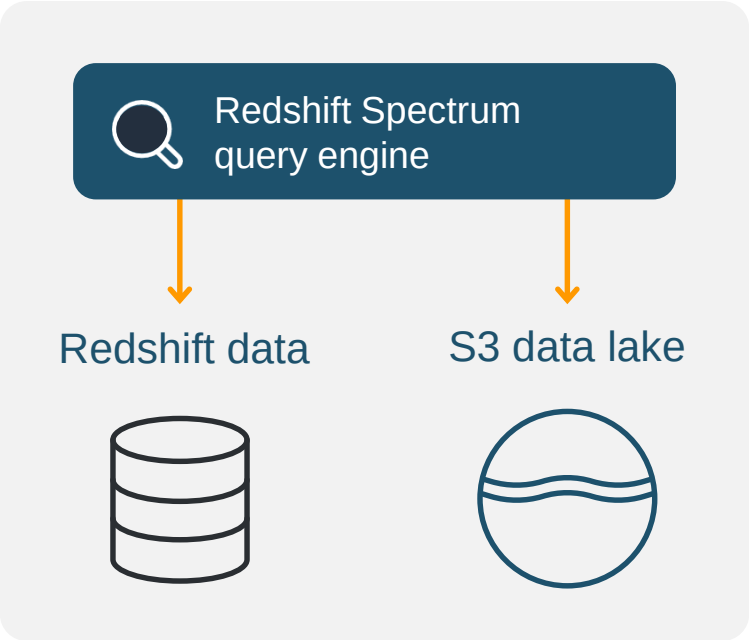
If you're using DC1 instances, upgrade to the DC2 instance type

- Same price as DC1, significantly faster
- Reserved Instances do not automatically transfer over

aws

Redshift Spectrum

Extend the data warehouse to your S3 data lake

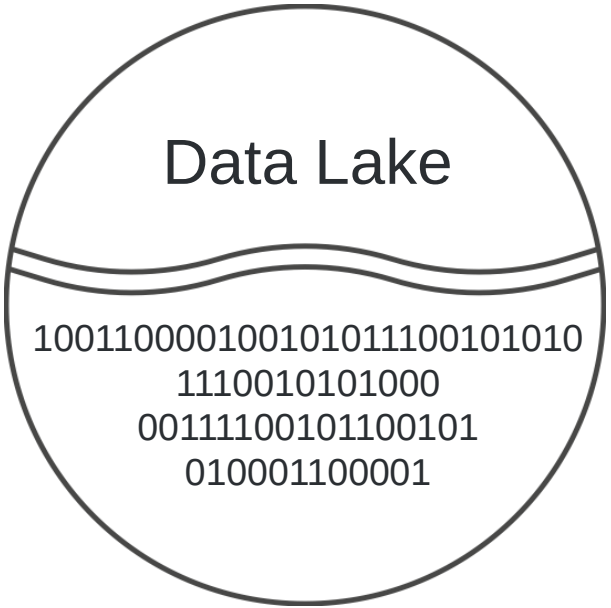


- Redshift SQL queries against exabytes in S3
- Join data across Redshift and S3
- Scale compute and storage separately
- Stable query performance and unlimited concurrency
- Parquet, ORC, Grok, Avro, & CSV data formats
- Pay only for the amount of data scanned

aws

Characteristics of a Data Lake

What is a Data Lake and what value does it provide



Collect and store any data, at any scale, and at low cost

Secure the data and prevent unauthorized access

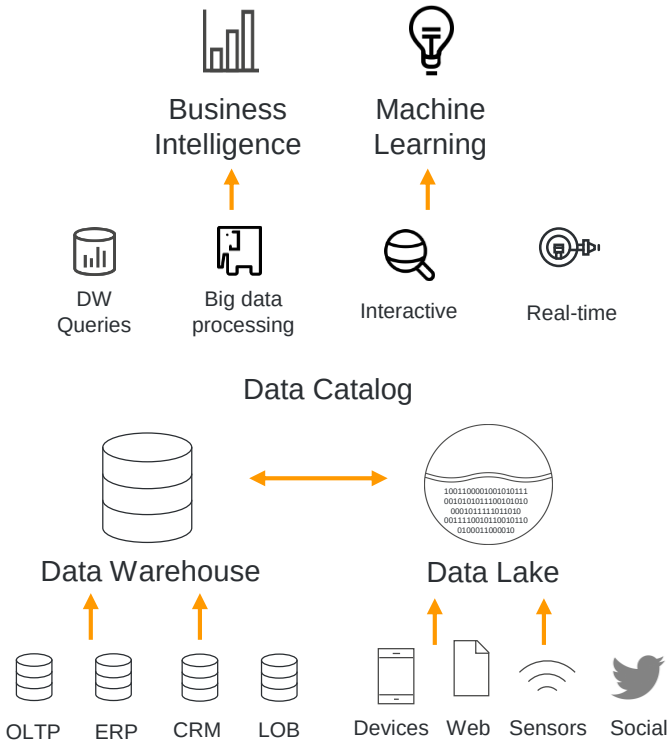
Catalogue, search, and discover data

Decouple Compute from Storage

Future Proof against a highly changing complex ecosystem

aws

Data Lakes extend traditional warehouses



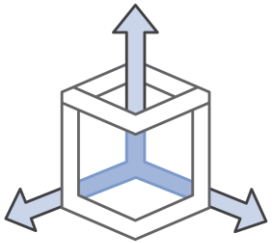
- Relational and non-relational data
- Terabytes to Exabytes scale
- Schema defined during analysis
- Diverse analytical engines to gain insights
- Designed for low cost storage and analytics

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Amazon Redshift Spectrum

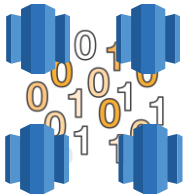
Query directly against data in Amazon S3 using thousands of nodes

Fast @ exabyte scale



Elastic & highly available

On-demand, pay-per-query



High concurrency: Multiple clusters access same data

Query data in-place using open file formats

Full Amazon Redshift SQL support

aws

Redshift Spectrum
Query your data lake

Query
SELECT COUNT(*)
FROM S3.EXT_TABLE
GROUP BY …

JDBC/ODBC

Amazon Redshift

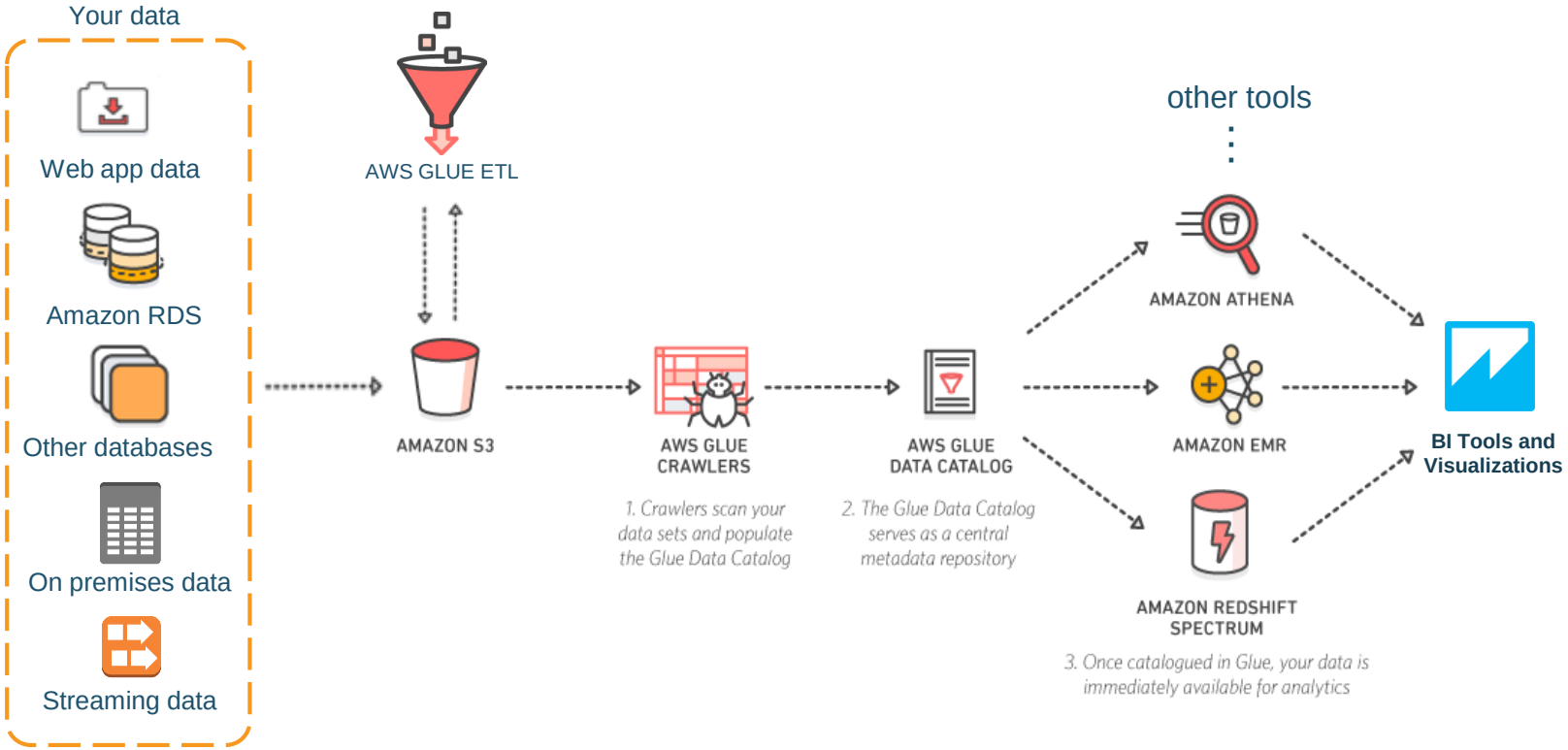
Redshift Spectrum
Scale-out serverless compute

1 2 3 4 … N

Amazon S3
Exabyte-scale object storage

AWS Glue
Data Catalog

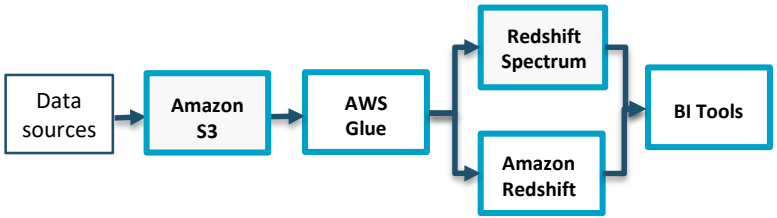
Data Lake on Amazon S3 with AWS Glue



© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

NUVIAD — Data Lake Analytics with Redshift Spectrum

NUVIAD is a marketing platform that helps media buyers optimize their mobile bidding



Use AWS for marketing campaign and bidding analytics

Scale S3 storage for unlimited data capacity

Use Spectrum for unlimited scale and query concurrency

80% performance gain using Parquet data format

*"**Amazon Redshift Spectrum is a game changer for us.** Reports that took minutes to produce are now delivered in seconds. We like the ability scale compute on-demand to query Petabytes of data in S3 in various open file formats."*

-- Rafi Ton, CEO, NUVIAD

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Let's build an analytic query - #1

An author is releasing the 8th book in her popular series. How many should we order for Seattle? What were prior first few day sales?

Let's compute the sales of the prior books she's written in this series and return the top 20 values

2 Tables (1 S3, 1 local)
2 Filters
1 Join
2 Group By columns
1 Order By
1 Limit
1 Aggregation

```
SELECT
       P.ASIN,
       P.TITLE,
       SUM(D.QUANTITY * D.OUR_PRICE) AS SALES_sum
FROM
       s3.d_customer_order_item_details D,
       products P
WHERE
       D.ASIN = P.ASIN AND
       P.TITLE LIKE '%Potter%' AND
       P.AUTHOR = 'J. K. Rowling' AND
GROUP BY P.ASIN, P.TITLE
ORDER BY SALES_sum DESC
LIMIT 20;
```


aws

Let's build an analytic query - #2

An author is releasing the 8th book in her popular series. How many should we order for Seattle? What were prior first few day sales?

Let's compute the sales of the prior books she's written in this series and return the top 20 values, just for the first three days of sales of first editions in the city of Seattle, WA, USA

4 Tables (1 S3, 3 local)
8 Filters
3 Joins
4 Group By columns
1 Order By
1 Limit
1 Aggregation
1 Function
2 Casts

```
SELECT
       P.ASIN,
       P.TITLE,
       R.POSTAL_CODE,
       P.RELEASE_DATE,
       SUM(D.QUANTITY * D.OUR_PRICE) AS SALES_sum
FROM
       s3.d_customer_order_item_details D,
       asin_attributes A,
       products P,
       regions R
WHERE
       D.ASIN = P.ASIN AND
       P.ASIN = A.ASIN AND
       D.REGION_ID = R.REGION_ID AND
       A.EDITION LIKE '%FIRST%' AND
       P.TITLE LIKE '%Potter%' AND
       P.AUTHOR = 'J. K. Rowling' AND
       R.COUNTRY_CODE = 'US' AND
       R.CITY = 'Seattle' AND
       R.STATE = 'WA' AND
       D.ORDER_DAY :: DATE >= P.RELEASE_DATE AND
       D.ORDER_DAY :: DATE < dateadd(day, 3, P.RELEASE_DATE)
GROUP BY P.ASIN, P.TITLE, R.POSTAL_CODE, P.RELEASE_DATE
ORDER BY SALES_sum DESC
LIMIT 20;
```


aws

Now let's run that query over an exabyte of data in S3

```
demo=# SELECT
demo-#   P.ASIN,
demo-#   P.TITLE,
demo-#   R.POSTAL_CODE,
demo-#   P.RELEASE_DATE,
demo-#   SUM(D.QUANTITY * D.OUR_PRICE) AS SALES_sum
demo-# FROM s3.d_customer_order_item_details D, asin_attributes A, products P, regions r
demo-# WHERE D.ASIN = P.ASIN AND
demo-#       P.ASIN = A.ASIN AND
demo-#       D.REGION_ID = R.REGION_ID AND
demo-#       A.EDITION LIKE '%FIRST%' AND
demo-#       P.TITLE LIKE '%Potter%' AND
demo-#       P.AUTHOR = 'J. K. Rowling' AND
demo-#       R.COUNTRY_CODE = 'US' AND
demo-#       R.CITY = 'Seattle' AND
demo-#       R.STATE = 'WA' AND
demo-#       D.ORDER_DAY :: DATE >= P.RELEASE_DATE AND
demo-#       D.ORDER_DAY :: DATE < dateadd(day, 3, P.RELEASE_DATE)
demo-# GROUP BY P.ASIN, P.TITLE, R.POSTAL_CODE, P.RELEASE_DATE
demo-# ORDER BY sales_sum DESC
demo-# LIMIT 20;
```

Roughly 140 TB of customer item order detail records for each day over past 20 years.

190 million files across 15,000 partitions in S3. One partition per day for USA and rest of world.

Need a **billion-fold** reduction in data processed.

Running this query using a 1000 node Hive cluster would take over 5 years.*

- Compression ……..……..5X
- Columnar file format……..……10X

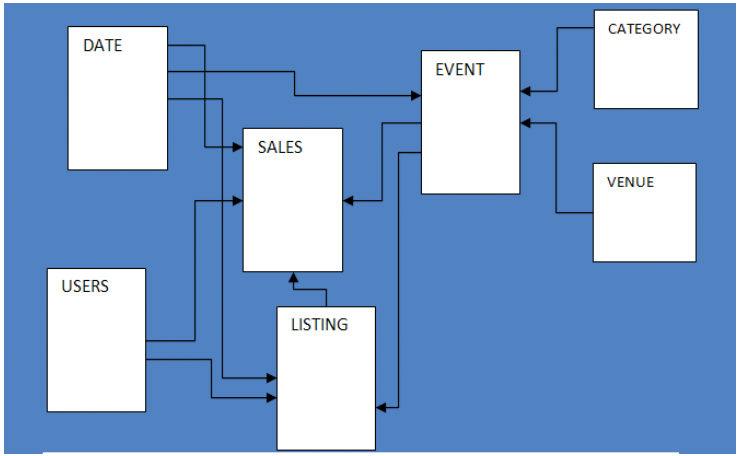
- Scanning with 2500 nodes…....2500X

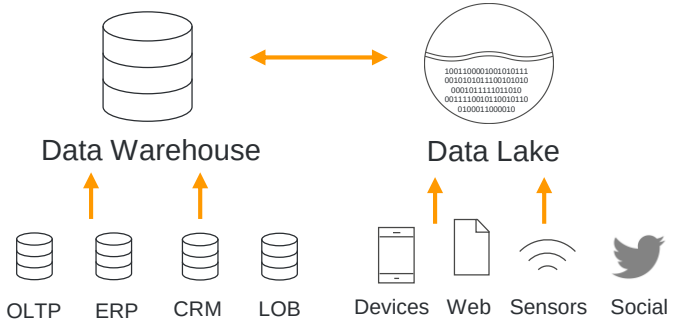
- Static partition elimination…..…….2X
- Dynamic partition elimination……350X
- Redshift's query optimizer……....40X

Total reduction……..……**3.5B X**

* Estimated using 20 node Hive cluster & 1.4TB, assume linear
* Query used a 20 node DC1.8XLarge Amazon Redshift cluster
* Not actual sales data - generated for this demo based on data format used by Amazon Retail.

Demonstration





Recommendations

Optimize your warehouse

Amazon Redshift Spectrum to improve scan-intensive concurrent workloads

Query Optimize your Data Lake

Use Column Oriented Storage on S3

Partition your objects

Redshift Spectrum Optimize

Know your query/node level parallelism

Improve Query performance

predicate pushdown

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Data Lakes on AWS

Building Big Data Storage Solutions (Data Lakes) for Maximum Flexibility - https://d1.awsstatic.com/whitepapers/Storage/data-lake-on-aws.pdf

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Thank you!