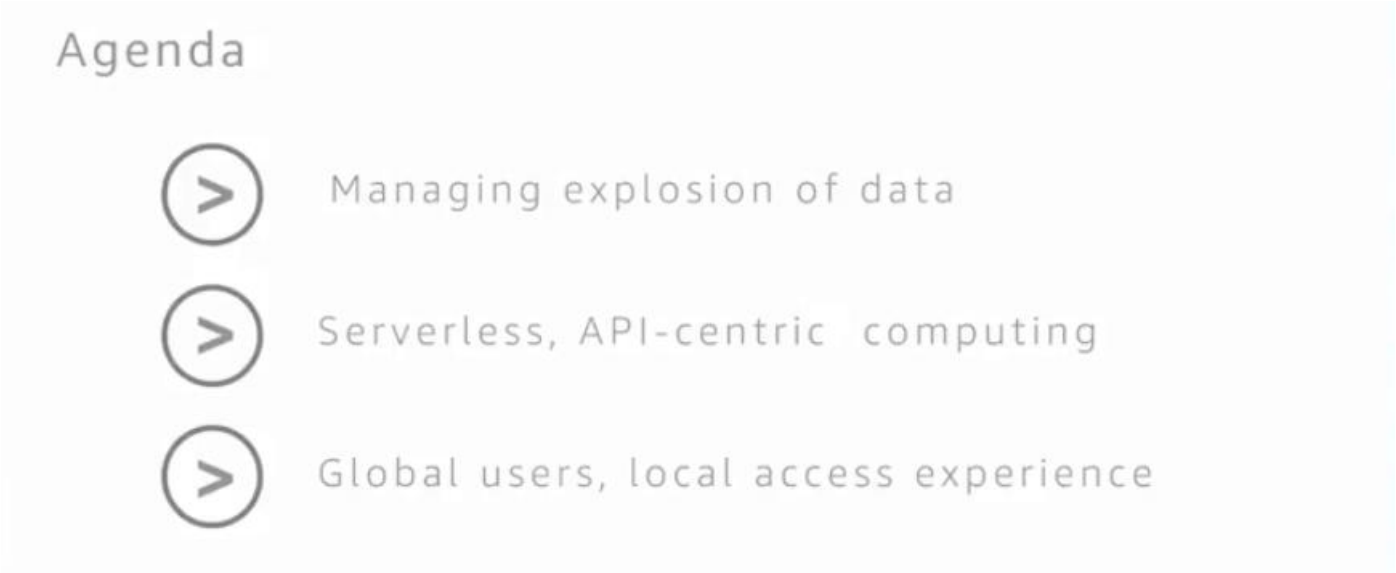
AWS re:INVENT

DAT311: Technology trends in Data Processing

Anurag Gupta, Vice President, Amazon Web Services
awgupta@amazon.com

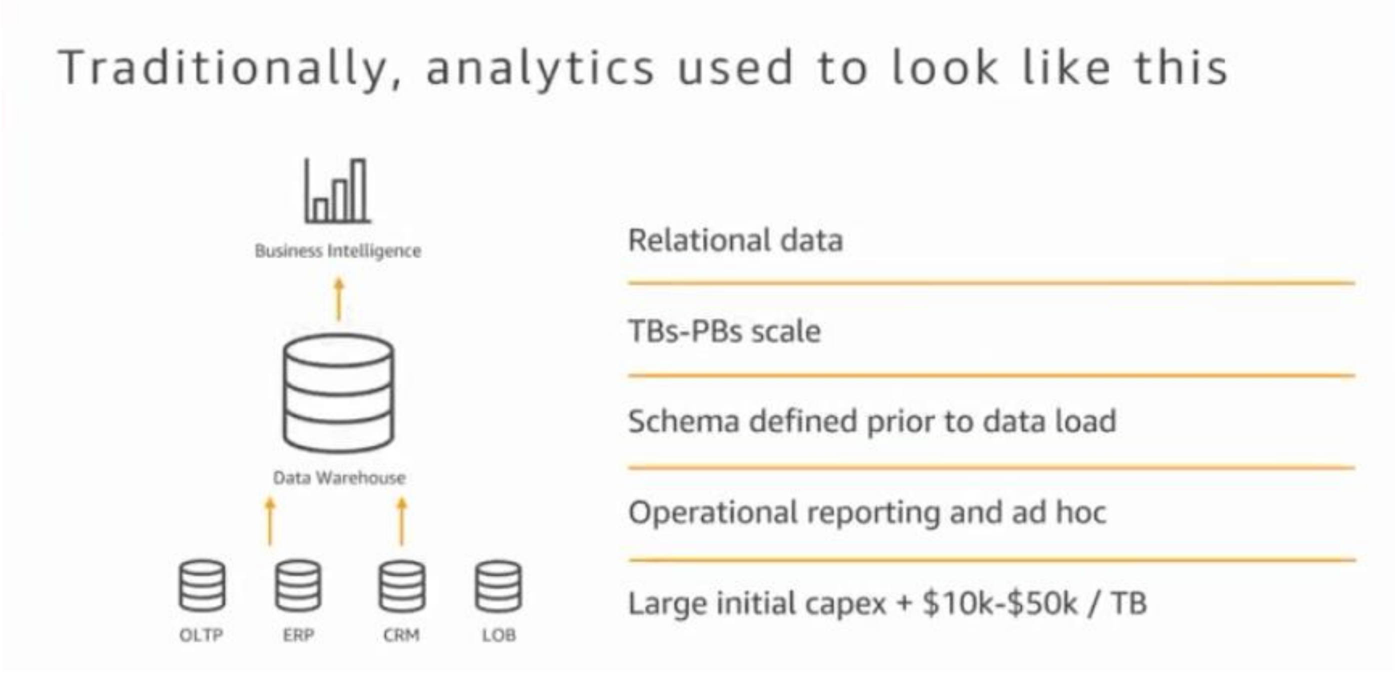
In this talk, Anurag Gupta, VP for AWS Analytic and Transactional Database Services, will talk about some of the key trends we see in data processing and how they shape the services we offer at AWS. Specific trends will include the rise of machine generated logs as the dominant source of data, the move towards Serverless, API-centric computing, and the growing need for local access to data from users around the world.



Agenda

- Managing explosion of data
- Serverless, API-centric computing
- Global users, local access experience

Managing Data Explosion with Data Lakes



Traditionally, analytics used to look like this

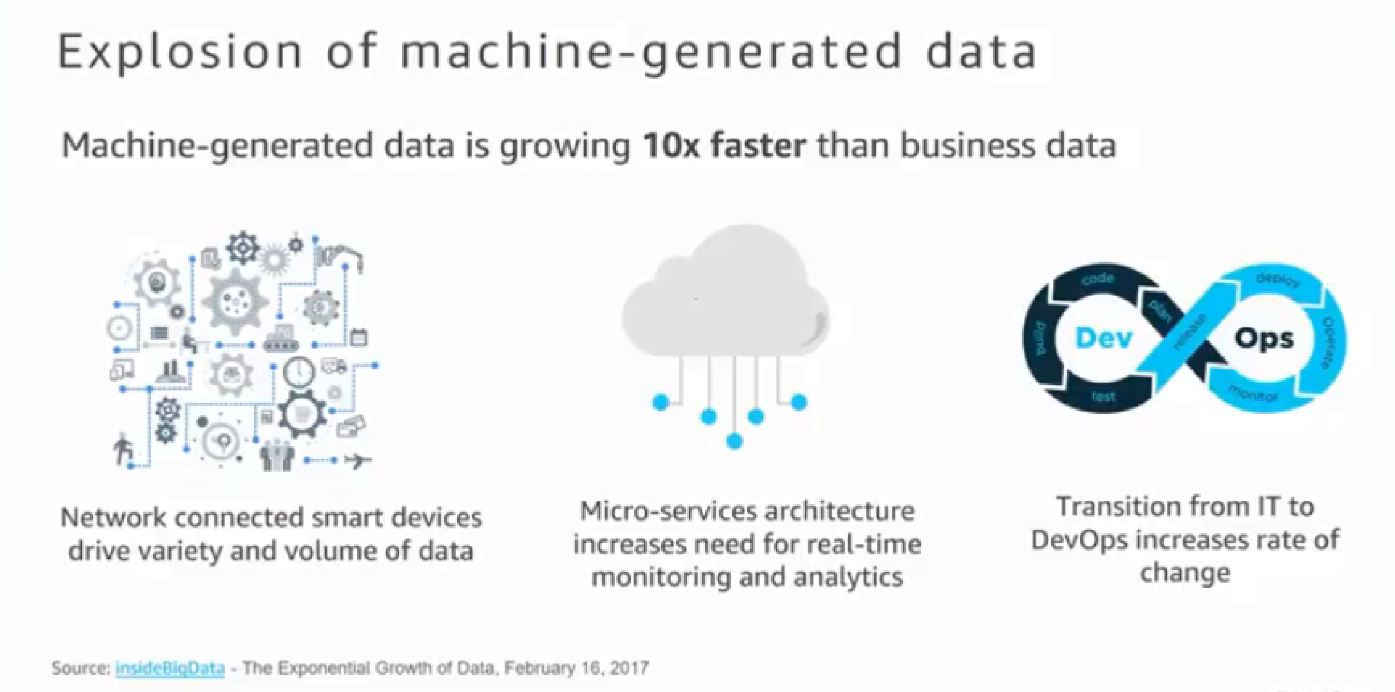
Relational data

TBs-PBs scale

Schema defined prior to data load

Operational reporting and ad hoc

Large initial capex + $10k-$50k / TB



Explosion of machine-generated data

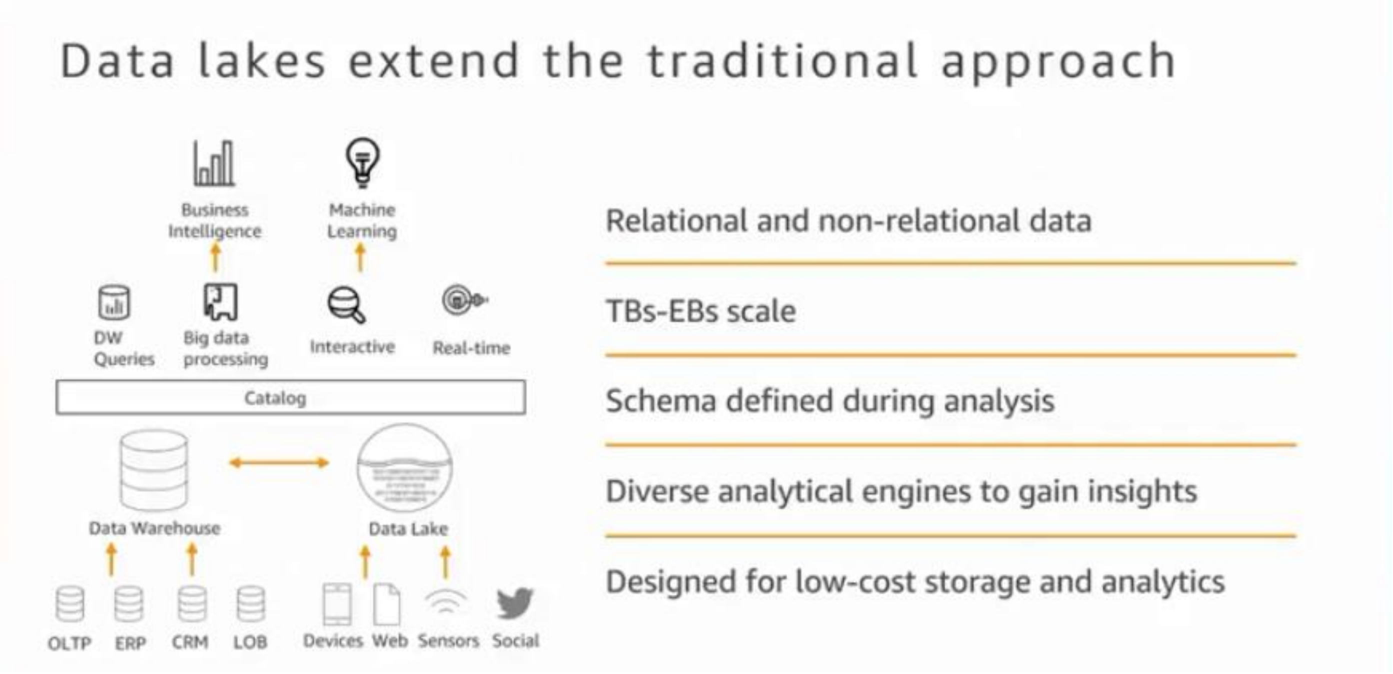
Machine-generated data is growing **10x faster** than business data

Network connected smart devices drive variety and volume of data

Micro-services architecture increases need for real-time monitoring and analytics

Transition from IT to DevOps increases rate of change

Source: insideBigData - The Exponential Growth of Data, February 16, 2017



Data lakes extend the traditional approach

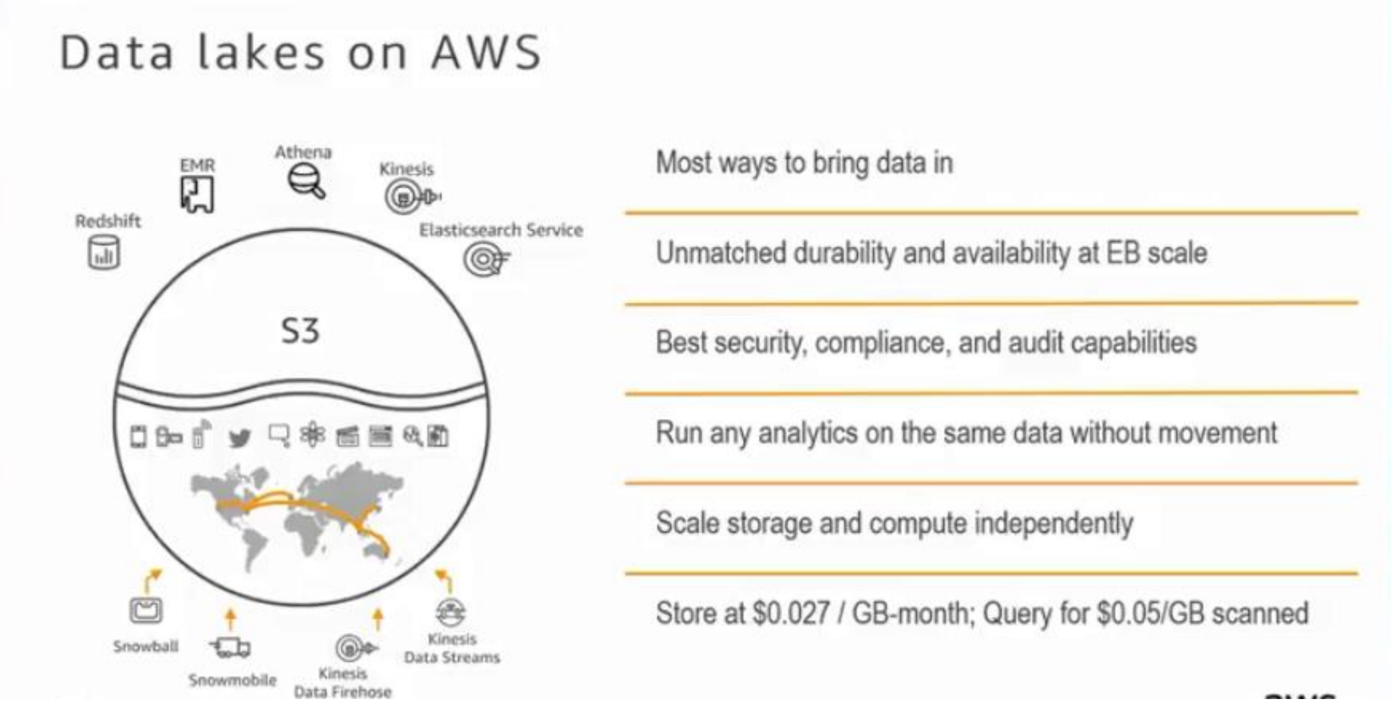
Relational and non-relational data

TBs-EBs scale

Schema defined during analysis

Diverse analytical engines to gain insights

Designed for low-cost storage and analytics



Data lakes on AWS

Most ways to bring data in

Unmatched durability and availability at EB scale

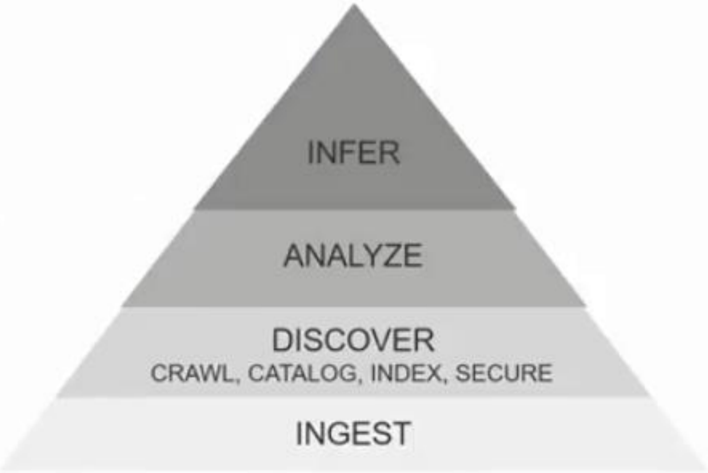
Best security, compliance, and audit capabilities

Run any analytics on the same data without movement

Scale storage and compute independently

Store at $0.027 / GB-month; Query for $0.05/GB scanned

Layers of a data lake



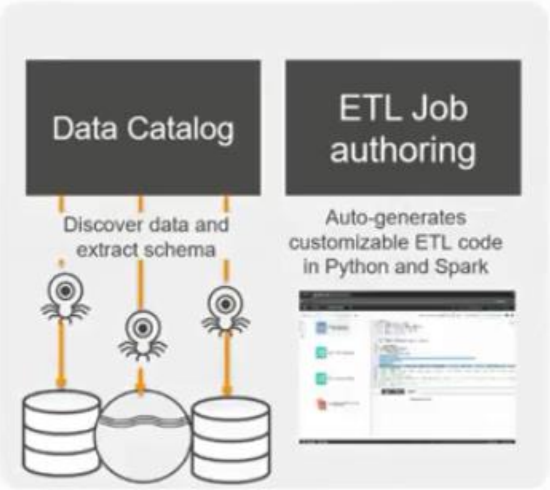
INFER

ANALYZE

DISCOVER
CRAWL, CATALOG, INDEX, SECURE

INGEST

AWS Glue – Serverless Data catalog & ETL service



Data Catalog

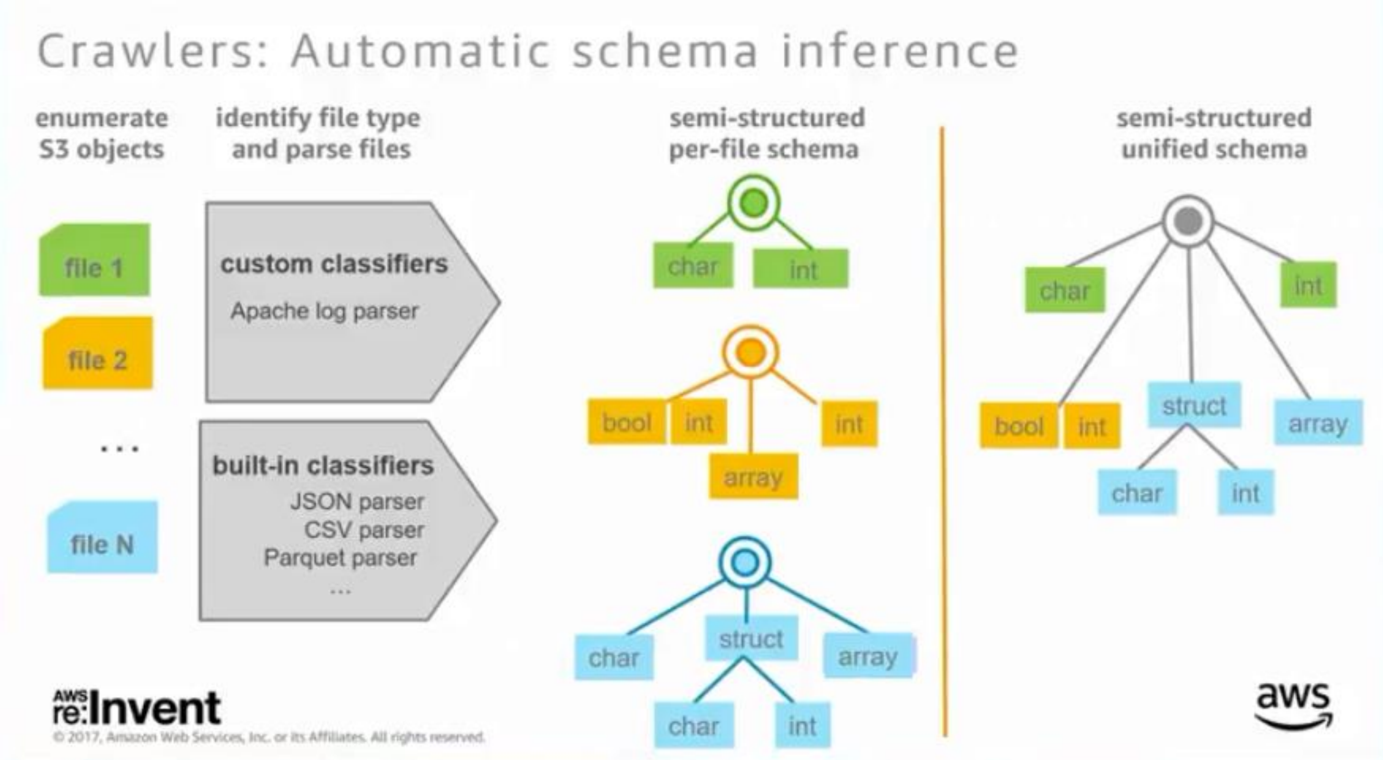
ETL Job authoring

Discover data and extract schema

Auto-generates customizable ETL code in Python and Spark

- Automatically discovers data and stores schema
- Data searchable, and available for ETL
- Generates customizable code
- Schedules and runs your ETL jobs
- Serverless

Crawlers: Automatic schema inference



enumerate S3 objects

identify file type and parse files

semi-structured per-file schema

semi-structured unified schema

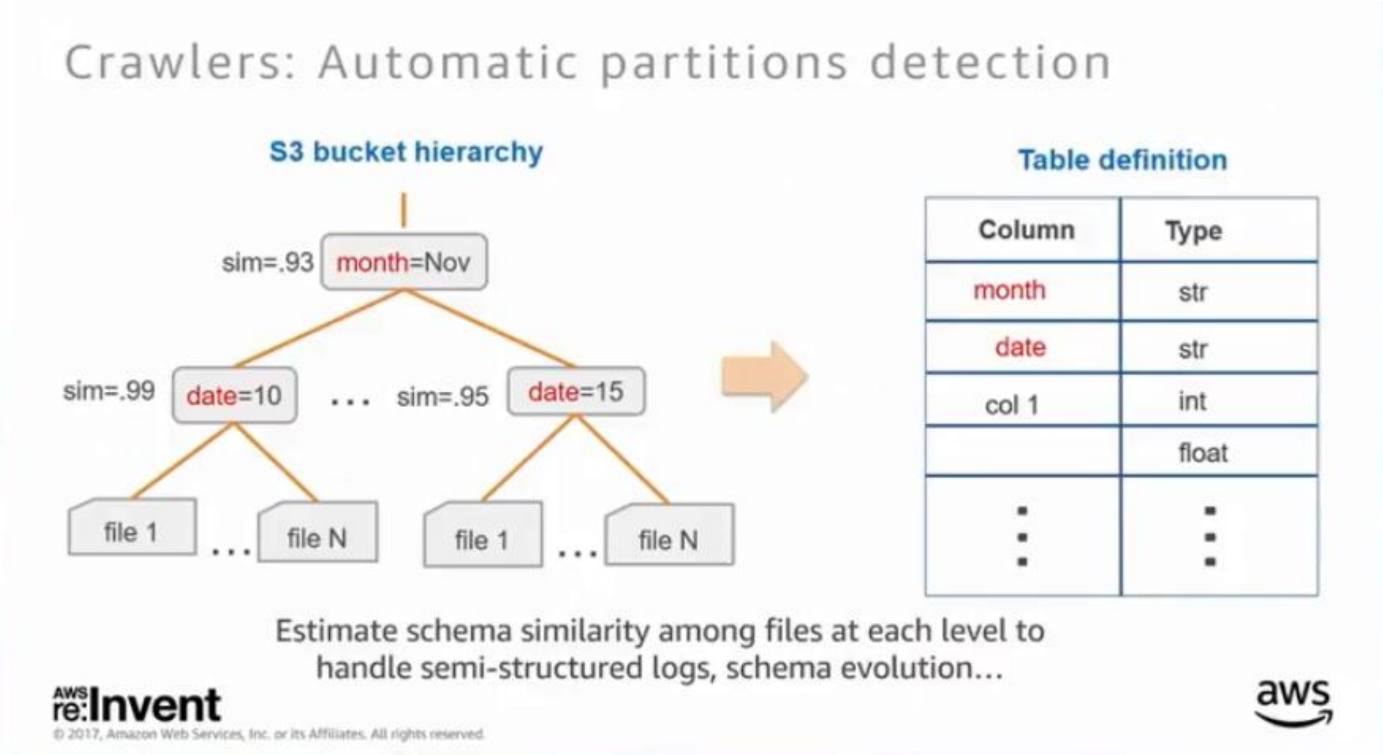
custom classifiers

Apache log parser

built-in classifiers

JSON parser
CSV parser
Parquet parser
...

Crawlers: Automatic partitions detection

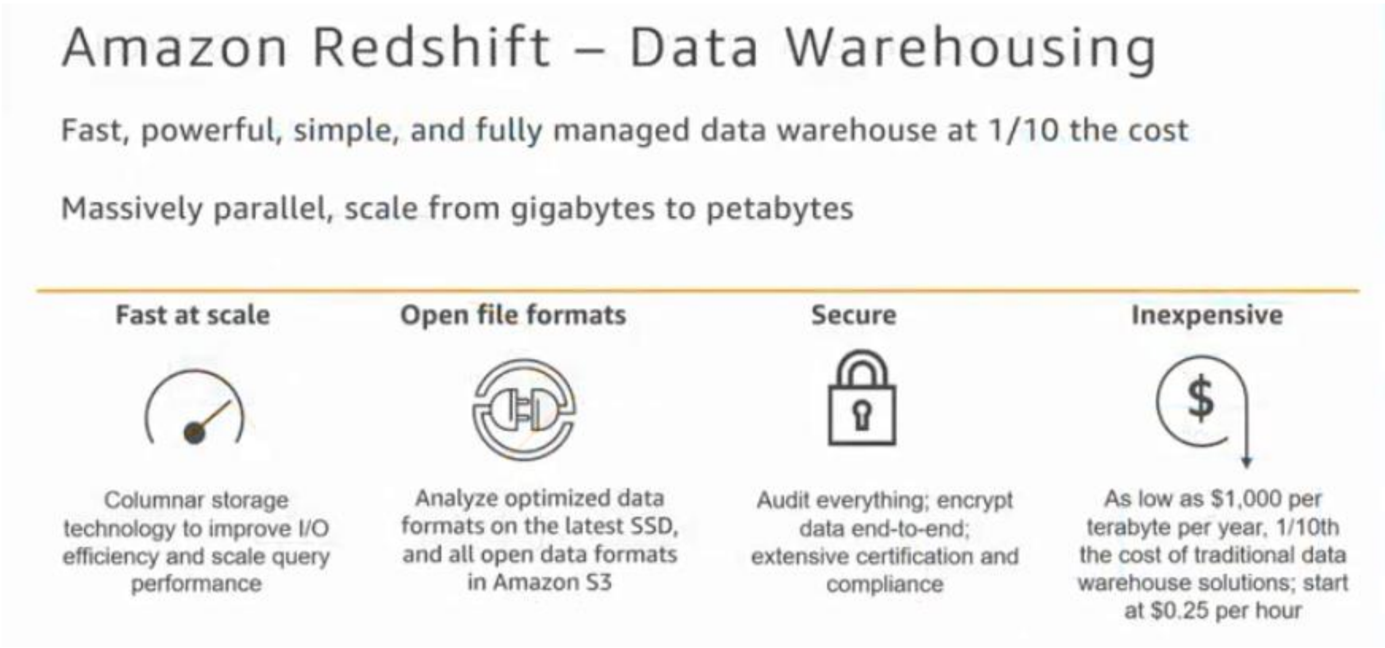


S3 bucket hierarchy

Table definition

Column	Type
month	str
date	str
col 1	int
	float
⋮	⋮

Estimate schema similarity among files at each level to handle semi-structured logs, schema evolution…



Amazon Redshift – Data Warehousing

Fast, powerful, simple, and fully managed data warehouse at 1/10 the cost

Massively parallel, scale from gigabytes to petabytes

Fast at scale

Columnar storage technology to improve I/O efficiency and scale query performance

Open file formats

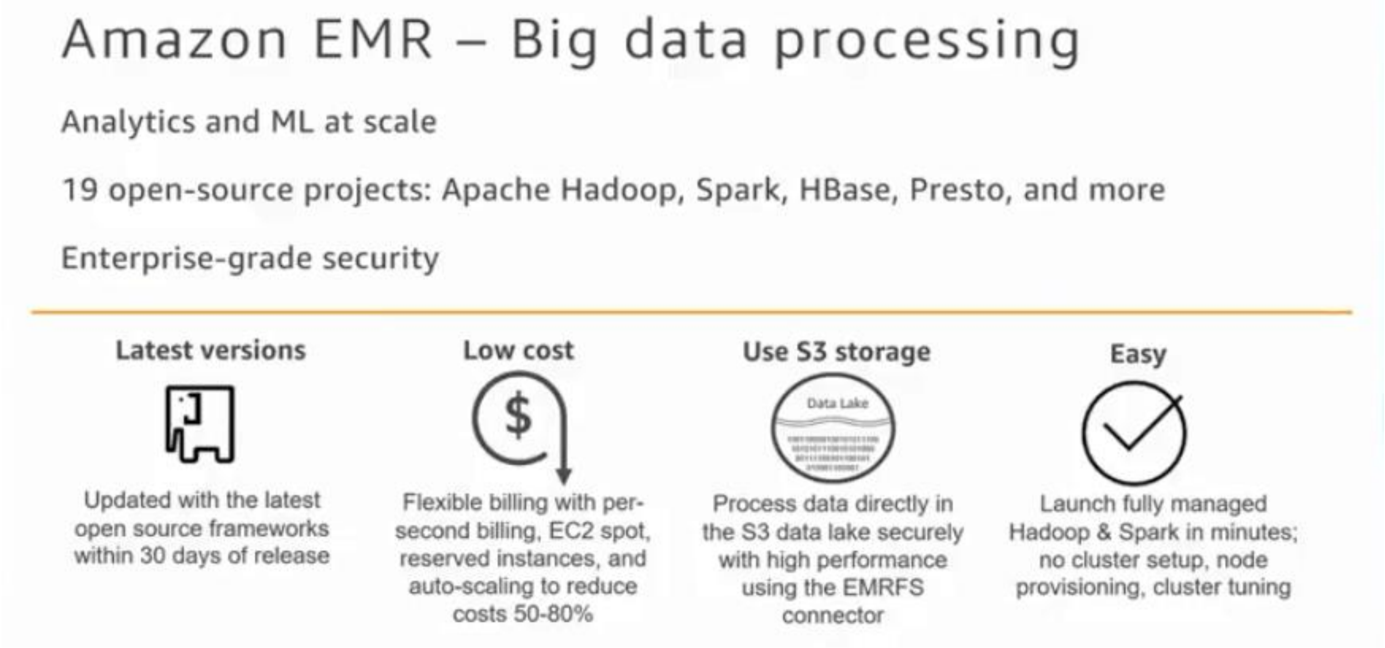
Analyze optimized data formats on the latest SSD, and all open data formats in Amazon S3

Secure

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

Inexpensive

As low as $1,000 per terabyte per year, 1/10th the cost of traditional data warehouse solutions; start at $0.25 per hour



Amazon EMR – Big data processing

Analytics and ML at scale

19 open-source projects: Apache Hadoop, Spark, HBase, Presto, and more

Enterprise-grade security

Latest versions

Updated with the latest open source frameworks within 30 days of release

Low cost

Flexible billing with per-second billing, EC2 spot, reserved instances, and auto-scaling to reduce costs 50-80%

Use S3 storage

Process data directly in the S3 data lake securely with high performance using the EMRFS connector

Easy

Launch fully managed Hadoop & Spark in minutes; no cluster setup, node provisioning, cluster tuning

Amazon Elasticsearch Service

Easy to deploy, secure, operate, and scale Elasticsearch

Customers use Elasticsearch for log analytics, full-text search, and application monitoring

Easy to Use

Fully-managed. Deploy production-ready clusters in minutes.

Open

Direct access to Elasticsearch open-source APIs. Supports Logstash and Kibana.

Secure

Secure access with VPC to keep all traffic within AWS network.

Available

Zone awareness replicates data between two AZs; automatically monitors and replaces failed nodes.

Amazon Kinesis – Real time

Easily to collect, process, and analyze video and data streams in real time

New

Kinesis Video Streams

Capture, process, and store video streams for analytics

Kinesis Data Streams

Build custom applications that analyze data streams

Kinesis Data Firehose

Load data streams into AWS data stores

Kinesis Data Analytics

Analyze data streams with SQL

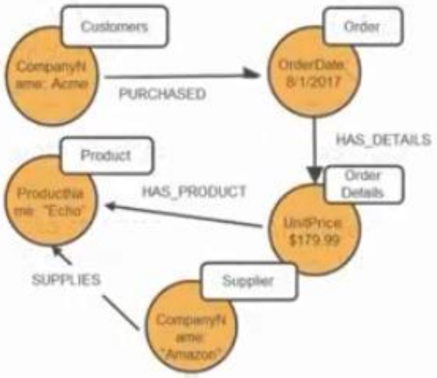
Highly connected data, best represented in a graph

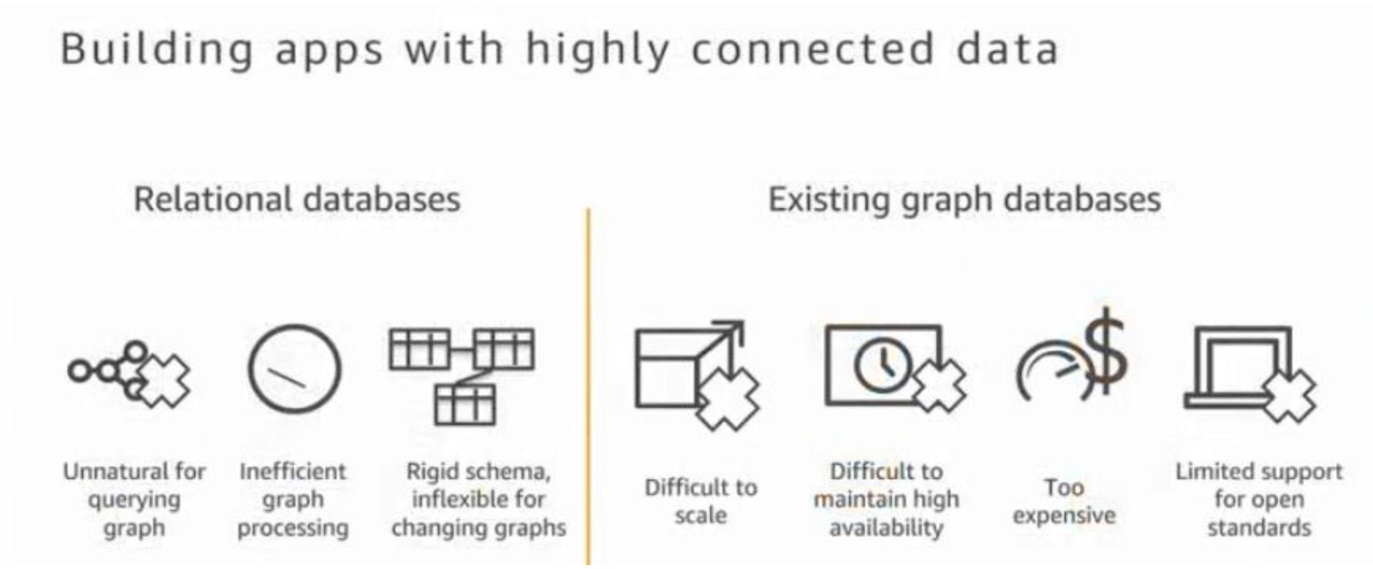
Relational model

Foreign keys used to represent relationships
Queries can involve nesting & complex joins
Performance can degrade as datasets grow

Graph model

Relationships are first-order citizens
Easy to write queries that navigate the graph
Results returned quickly, even on large datasets





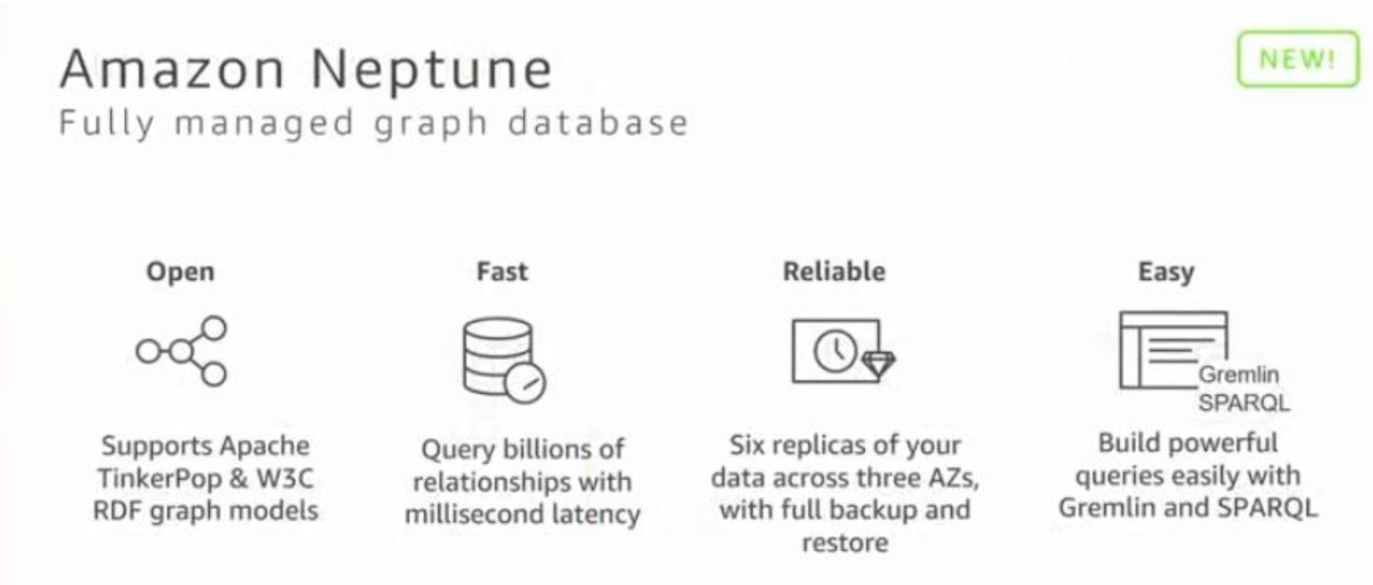
Building apps with highly connected data

Relational databases

- Unnatural for querying graph
- Inefficient graph processing
- Rigid schema, inflexible for changing graphs

Existing graph databases

- Difficult to scale
- Difficult to maintain high availability
- Too expensive
- Limited support for open standards



Amazon Neptune

Fully managed graph database

NEW!

Open

Supports Apache TinkerPop & W3C RDF graph models

Fast

Query billions of relationships with millisecond latency

Reliable

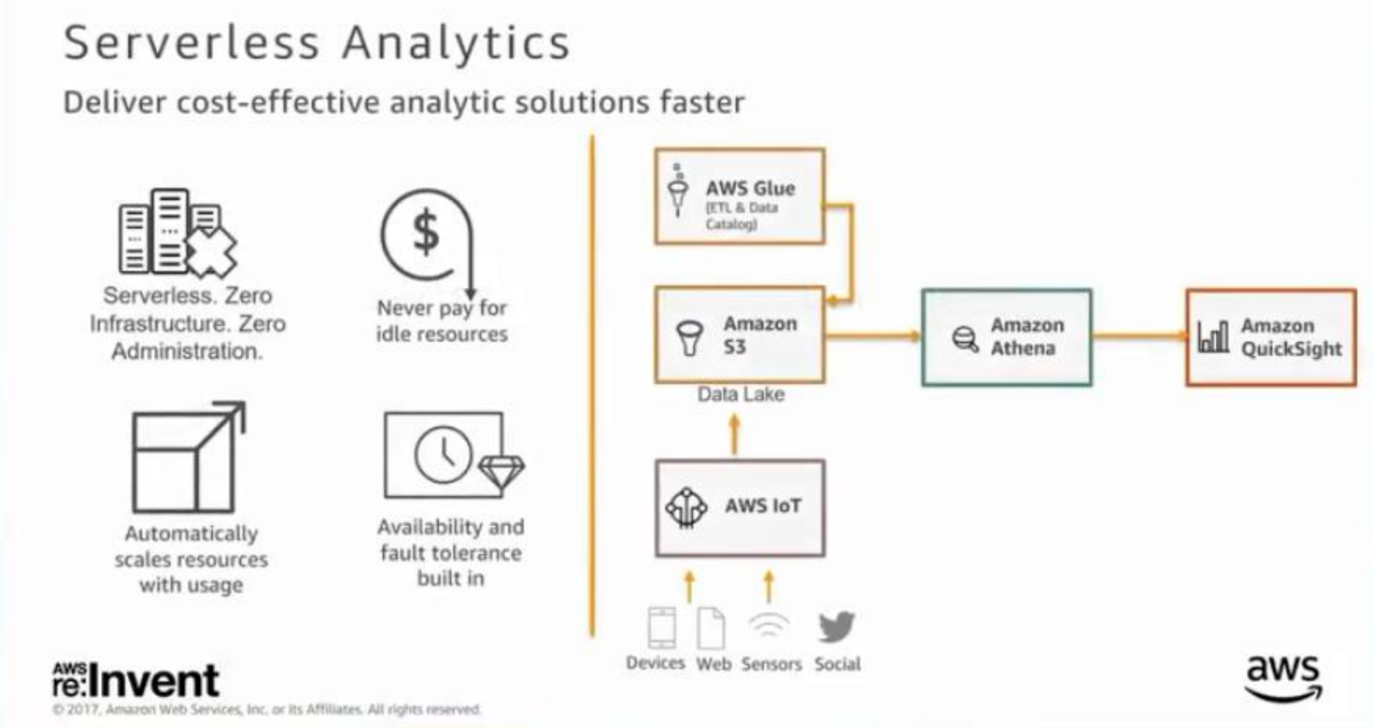
Six replicas of your data across three AZs, with full backup and restore

Easy

Gremlin
SPARQL

Build powerful queries easily with Gremlin and SPARQL

Serverless, API-Centric Computing



Serverless Analytics

Deliver cost-effective analytic solutions faster

Serverless. Zero Infrastructure. Zero Administration.

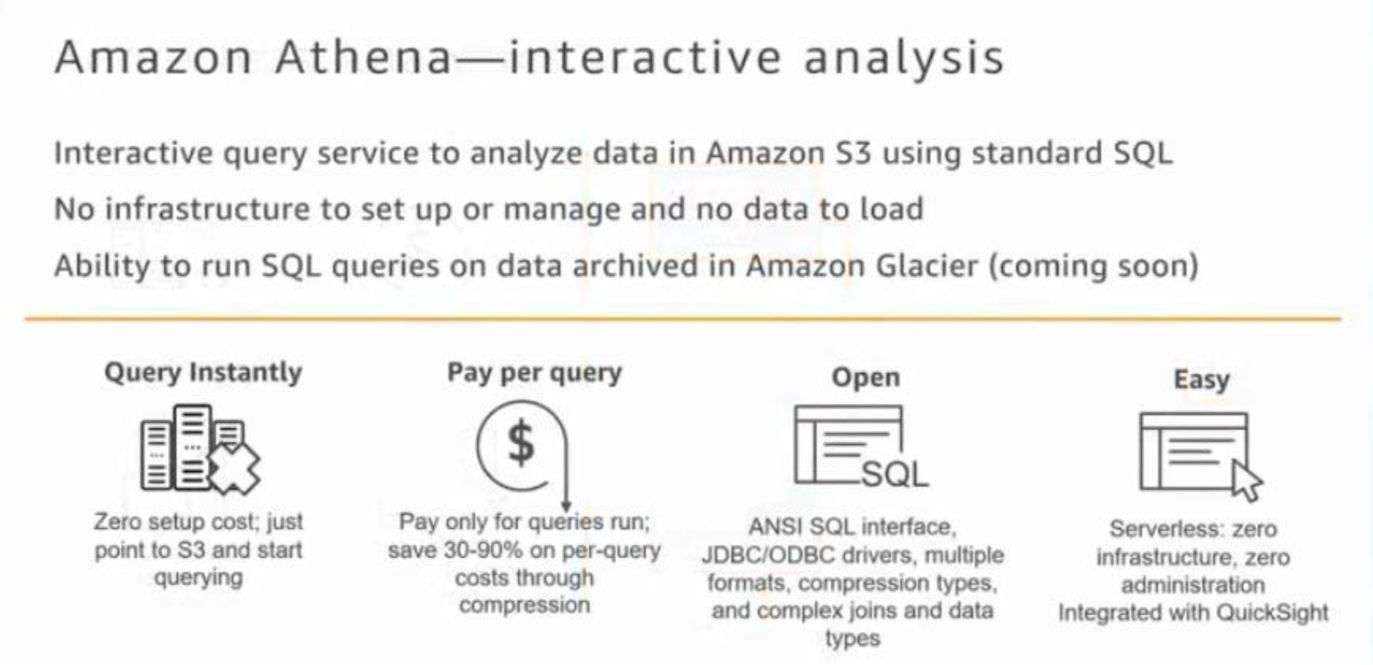
Never pay for idle resources

Automatically scales resources with usage

Availability and fault tolerance built in

re:Invent

aws



Amazon Athena—interactive analysis

Interactive query service to analyze data in Amazon S3 using standard SQL

No infrastructure to set up or manage and no data to load

Ability to run SQL queries on data archived in Amazon Glacier (coming soon)

Query Instantly

Zero setup cost; just point to S3 and start querying

Pay per query

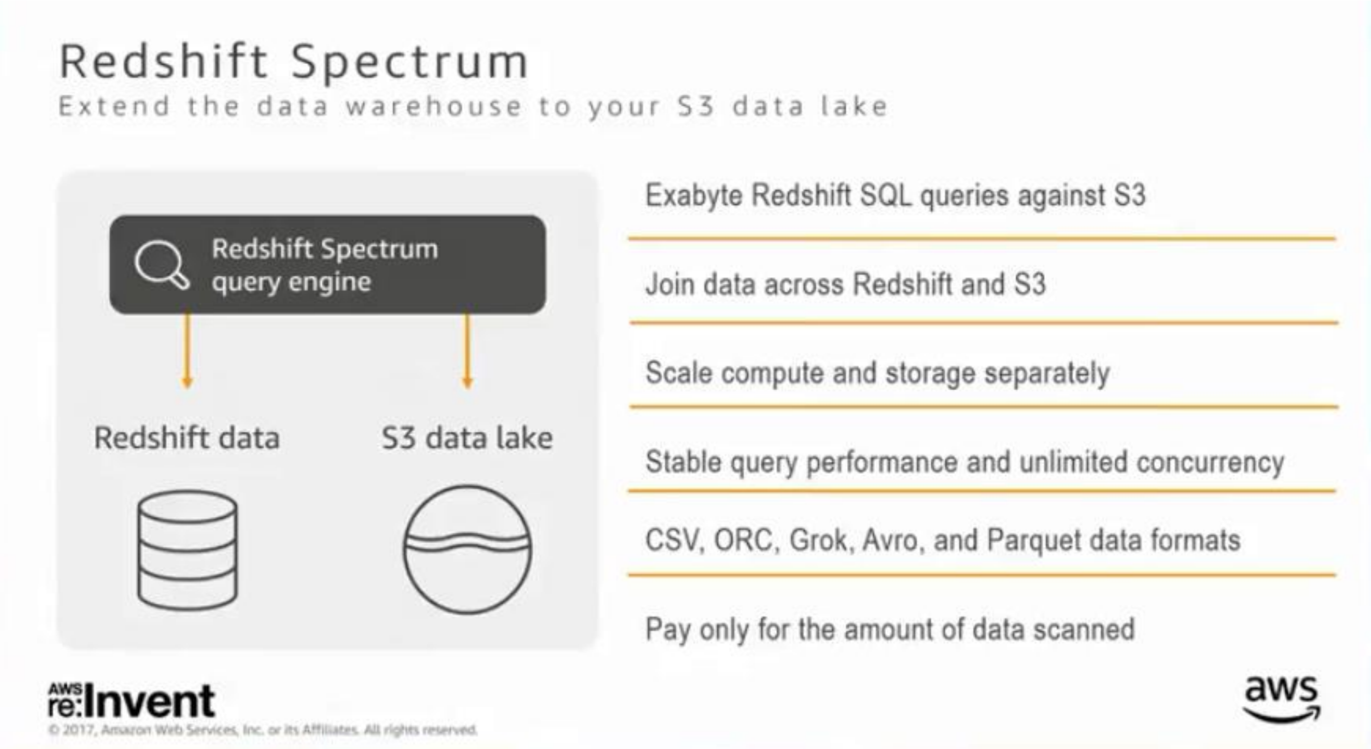
Pay only for queries run; save 30-90% on per-query costs through compression

Open

ANSI SQL interface, JDBC/ODBC drivers, multiple formats, compression types, and complex joins and data types

Easy

Serverless: zero infrastructure, zero administration
Integrated with QuickSight



Redshift Spectrum

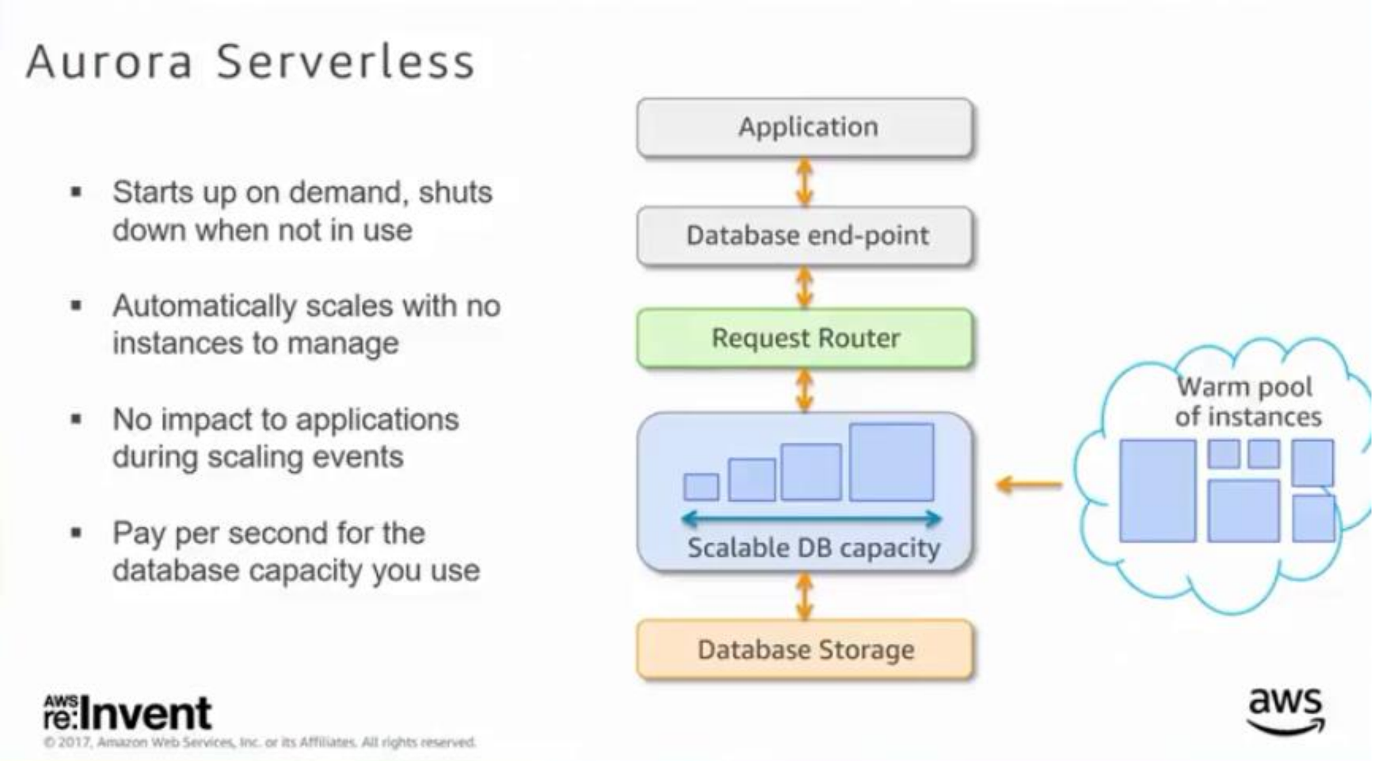
Extend the data warehouse to your S3 data lake

Redshift Spectrum query engine

Redshift data

S3 data lake

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



Aurora Serverless

- Starts up on demand, shuts down when not in use
- Automatically scales with no instances to manage
- No impact to applications during scaling events
- Pay per second for the database capacity you use

Application

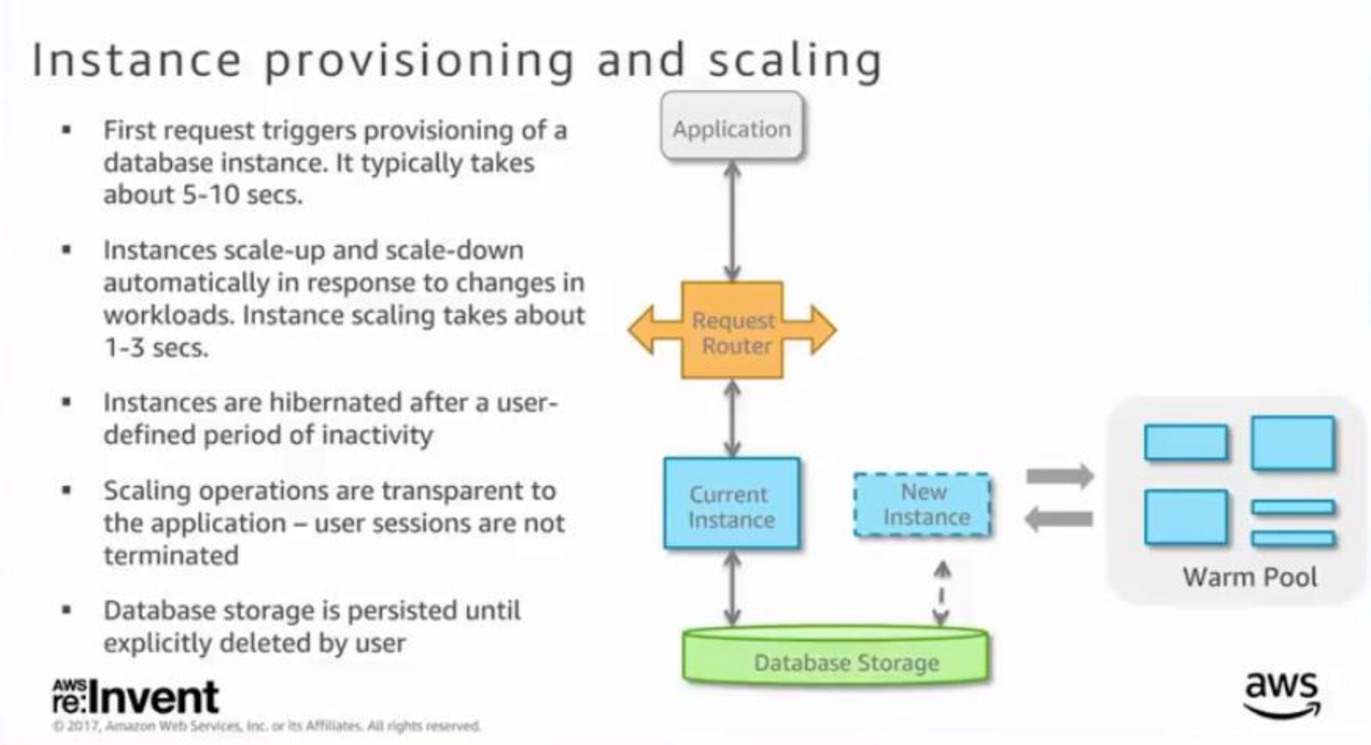
Database end-point

Request Router

Scalable DB capacity

Warm pool of instances

Database Storage



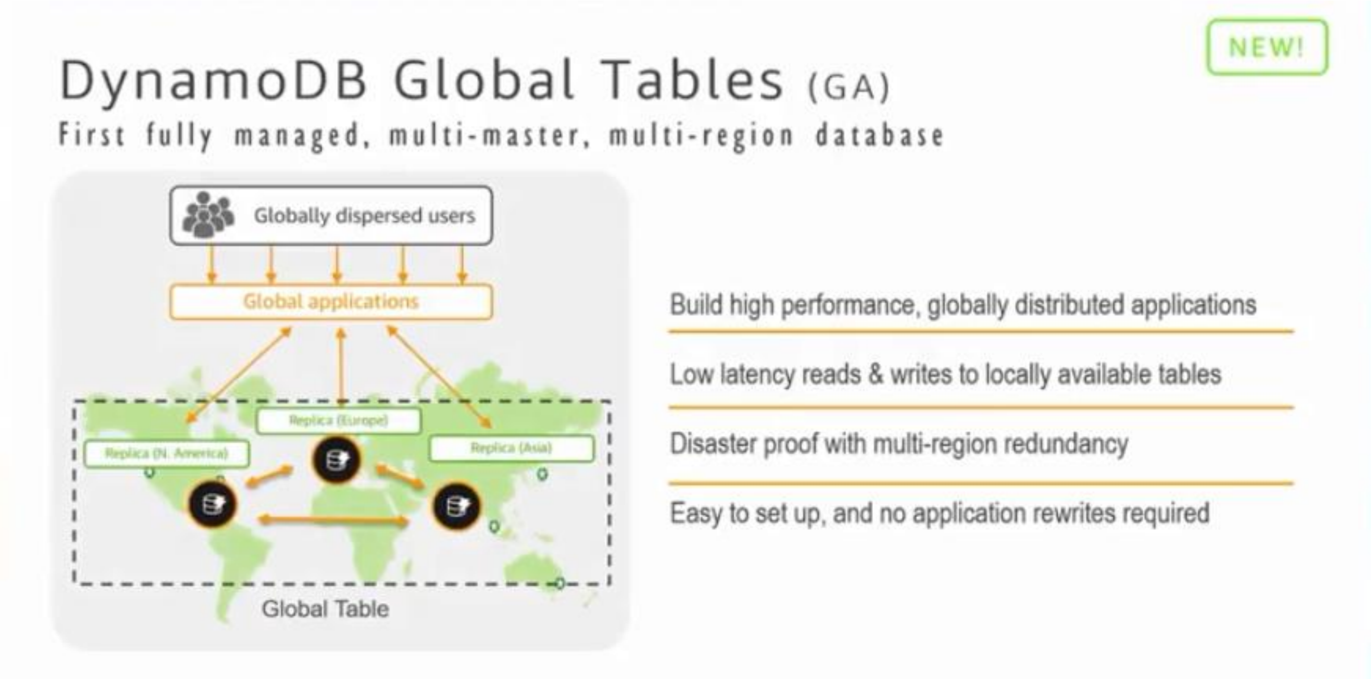
Instance provisioning and scaling

- First request triggers provisioning of a database instance. It typically takes about 5-10 secs.
- Instances scale-up and scale-down automatically in response to changes in workloads. Instance scaling takes about 1-3 secs.
- Instances are hibernated after a user-defined period of inactivity
- Scaling operations are transparent to the application – user sessions are not terminated
- Database storage is persisted until explicitly deleted by user

AWS re:Invent

aws

Global Users, Local Processing

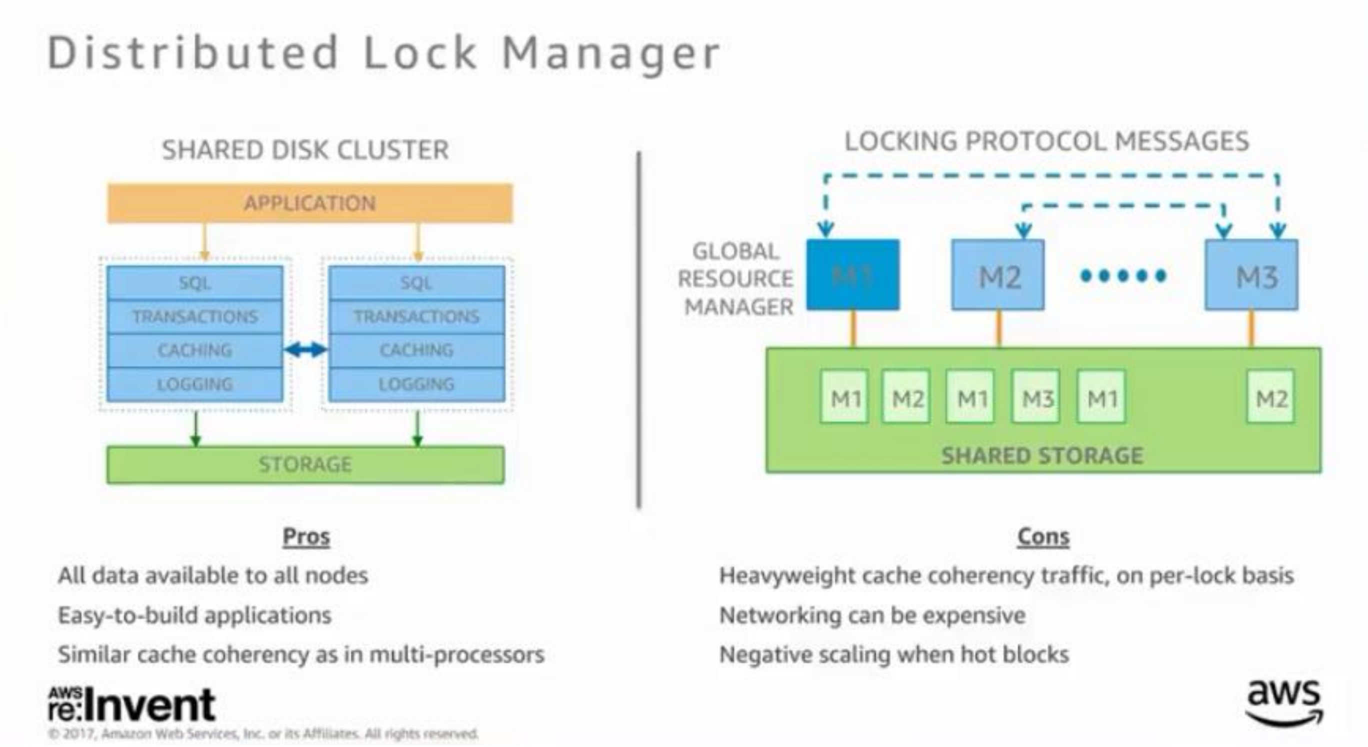


NEW!

DynamoDB Global Tables (GA)

First fully managed, multi-master, multi-region database

- Build high performance, globally distributed applications
- Low latency reads & writes to locally available tables
- Disaster proof with multi-region redundancy
- Easy to set up, and no application rewrites required



Distributed Lock Manager

Pros

All data available to all nodes

Easy-to-build applications

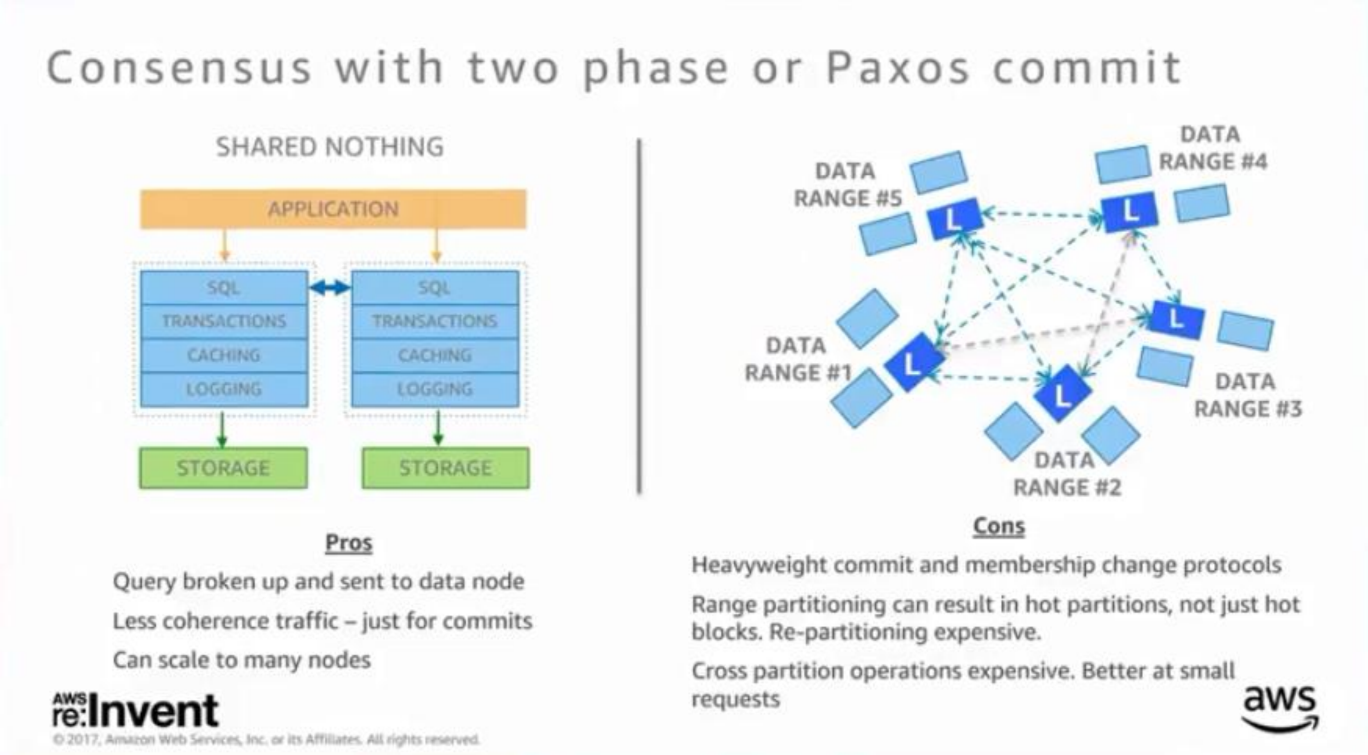
Similar cache coherency as in multi-processors

Cons

Heavyweight cache coherency traffic, on per-lock basis

Networking can be expensive

Negative scaling when hot blocks



Consensus with two phase or Paxos commit

Pros

Query broken up and sent to data node

Less coherence traffic – just for commits

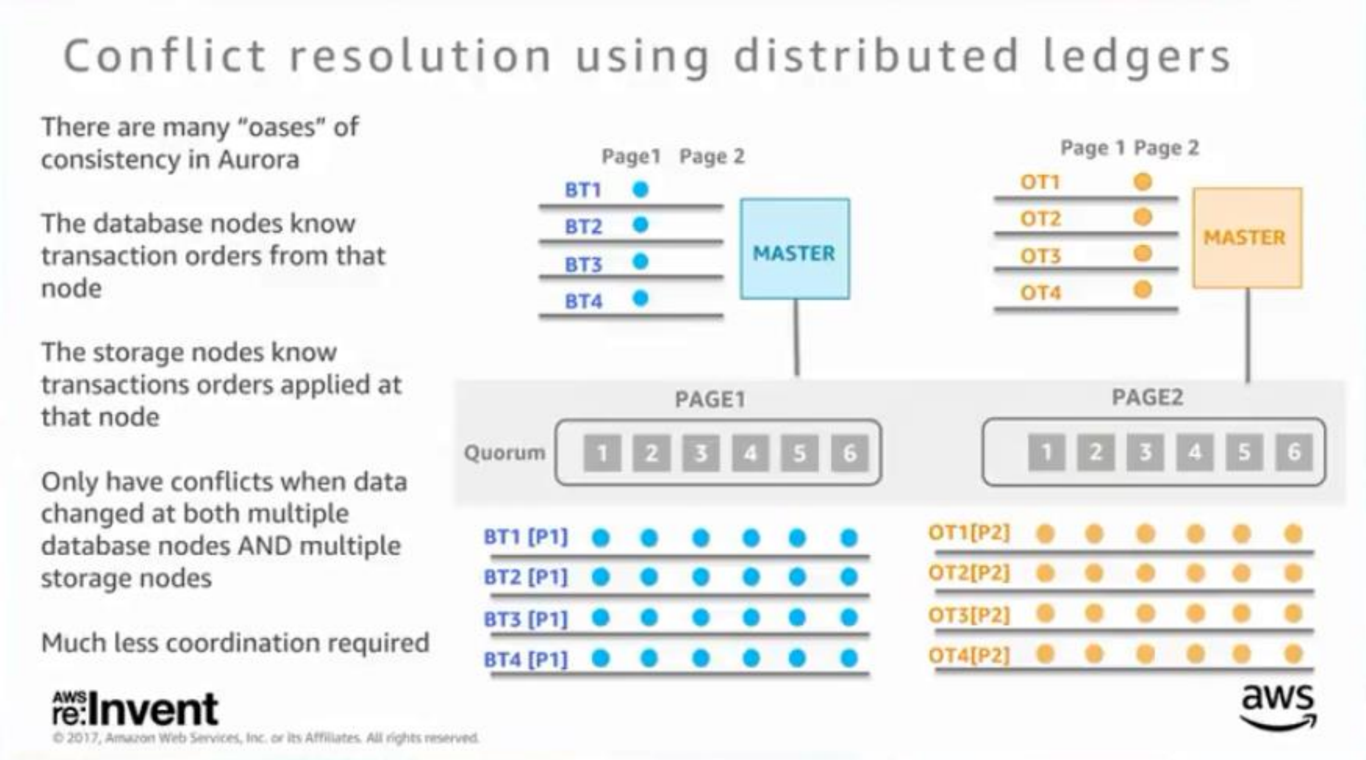
Can scale to many nodes

Cons

Heavyweight commit and membership change protocols

Range partitioning can result in hot partitions, not just hot blocks. Re-partitioning expensive.

Cross partition operations expensive. Better at small requests



Conflict resolution using distributed ledgers

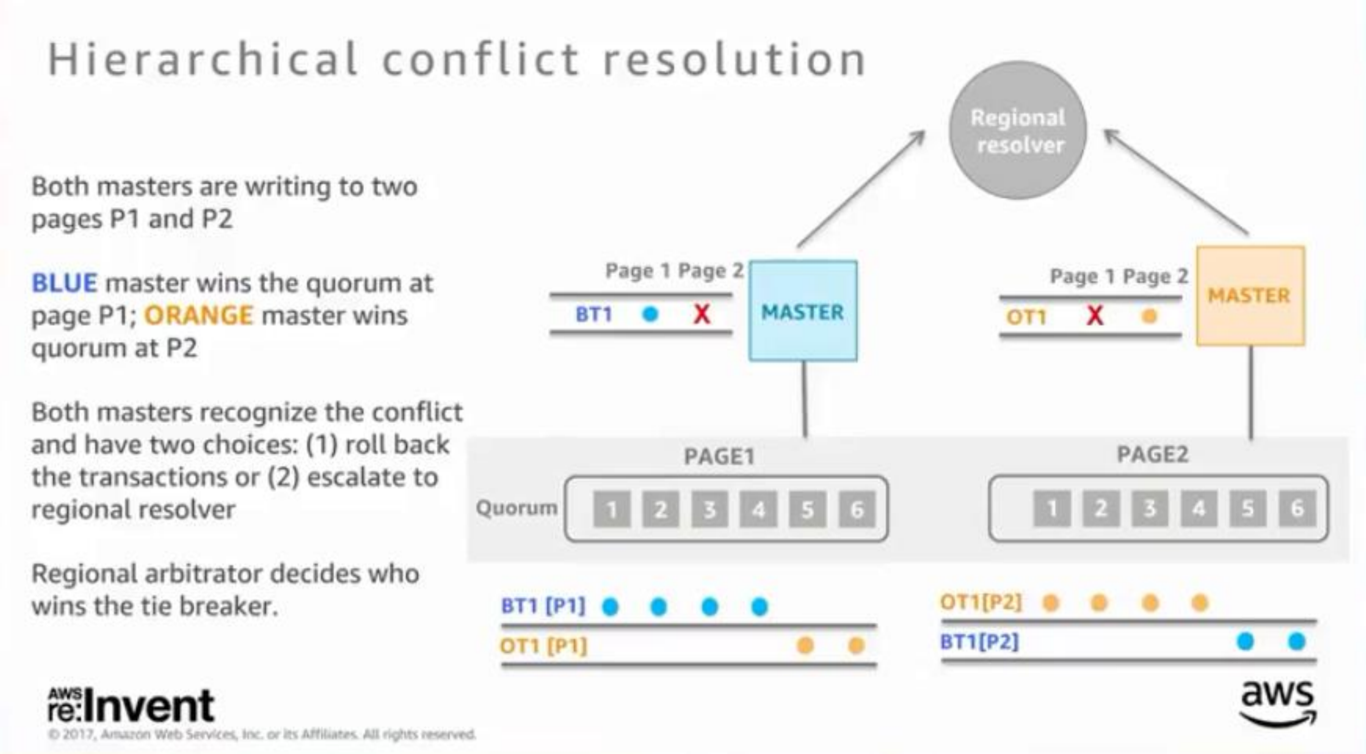
There are many "oases" of consistency in Aurora

The database nodes know transaction orders from that node

The storage nodes know transactions orders applied at that node

Only have conflicts when data changed at both multiple database nodes AND multiple storage nodes

Much less coordination required



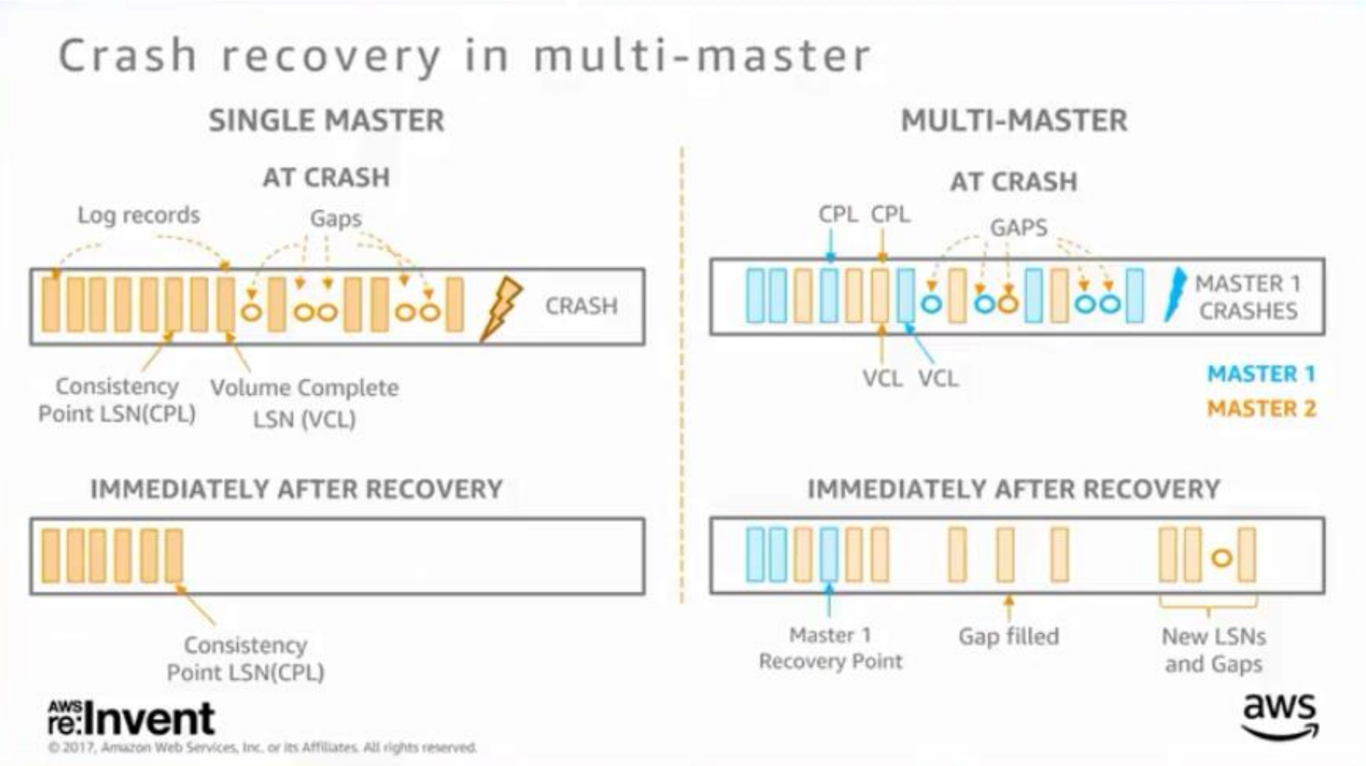
Hierarchical conflict resolution

Both masters are writing to two pages P1 and P2

BLUE master wins the quorum at page P1; **ORANGE** master wins quorum at P2

Both masters recognize the conflict and have two choices: (1) roll back the transactions or (2) escalate to regional resolver

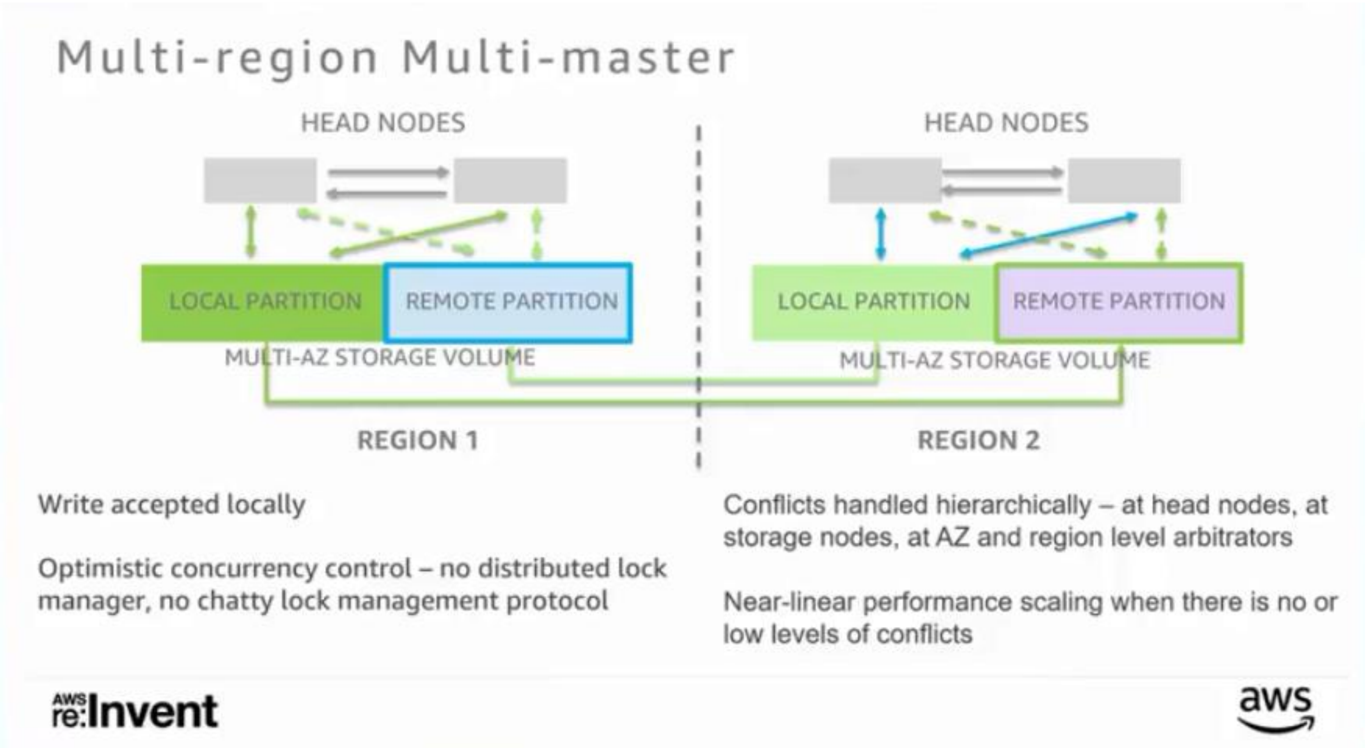
Regional arbitrator decides who wins the tie breaker.



Crash recovery in multi-master

AWS re:Invent

aws



Multi-region Multi-master

Write accepted locally

Optimistic concurrency control – no distributed lock manager, no chatty lock management protocol

Conflicts handled hierarchically – at head nodes, at storage nodes, at AZ and region level arbitrators

Near-linear performance scaling when there is no or low levels of conflicts

AWS re:Invent

aws

AWS re:Invent

Thank you!