Design and Evolution of Data Platforms

Matei Zaharia
CS 320

Stanford

Outline

History of information systems

Challenges in using data

Example solutions

New trends & Databricks case study

Outline

History of information systems

Challenges in using data

Example solutions

New trends & Databricks case study

Pre-Computer Era

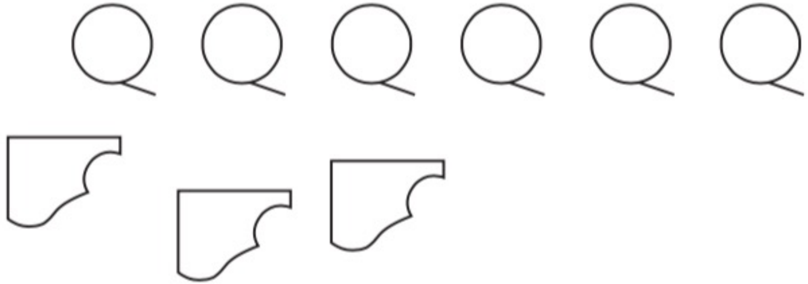
Early Business Computers

First used for **operational** applications (automate a business process)

- Bank’s account system
- Airline reservations
- Inventory

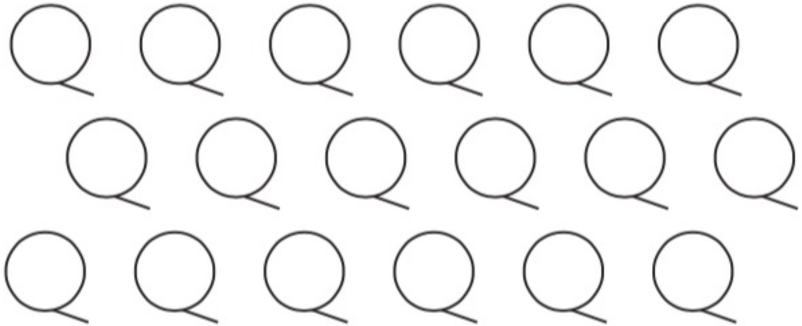
Each app manages own data (“master file”)

1960



master files, reports

1965



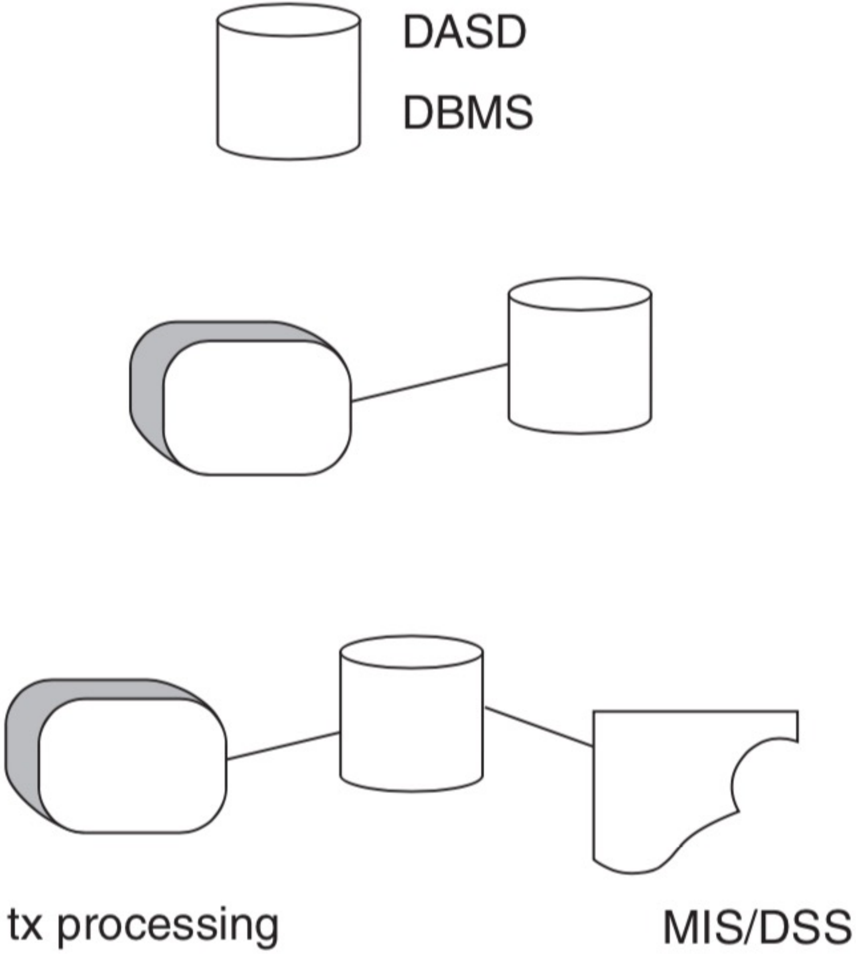
- complexity of—
  - maintenance
  - development
- synchronization of data
- hardware

lots of master files !!!

1970

1975

1980



database—"a single source of data for all processing"

IBM

online, high-performance transaction processing

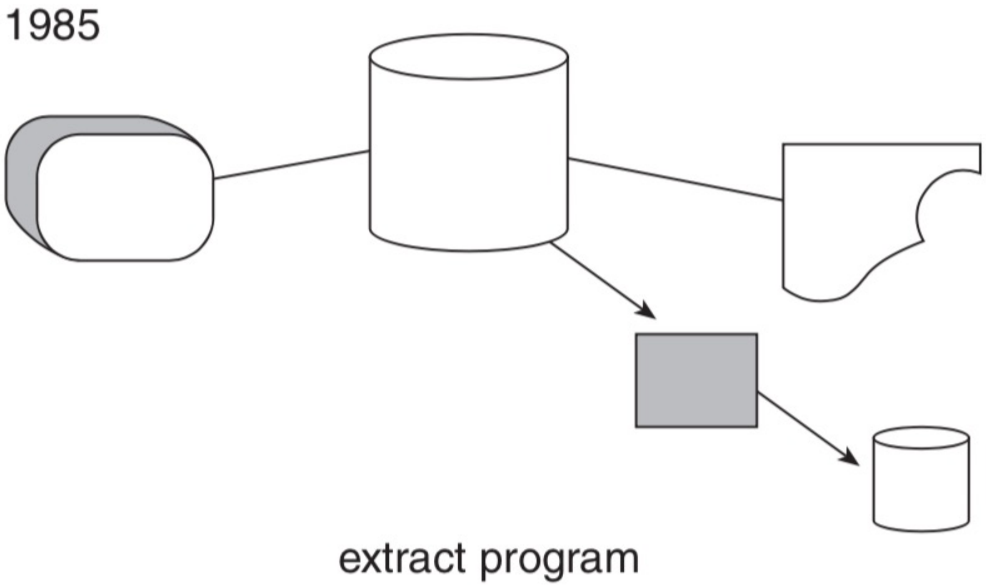
IBM

ORACLE

PCs, 4GL technology

SQL!

These databases are great for **operational** applications, but how to use them for **analytics**?


1985
extract program

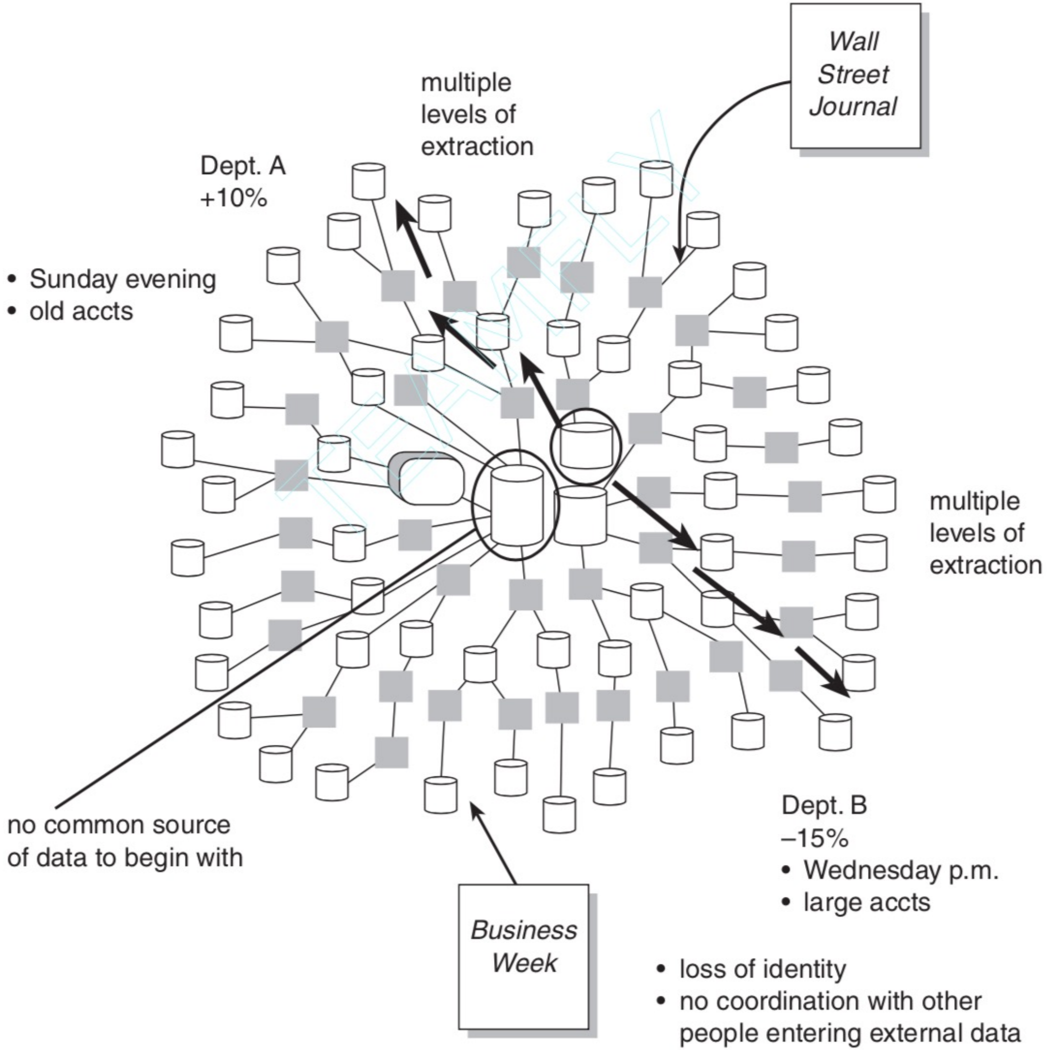
The Problem

Many systems with data about the same entities

Differing definitions of fields, identifiers, etc

Different points in time

Wrong / low quality data



Challenges in Using Data

These challenges are not unique to operational business apps, but occur in any modern data application / product

Core problem: when the amount of data is more than humans can review, it’s hard to tell whether it’s correct!

(Things get even harder with statistics and AI)

Solution: disciplines of data architecture & data engineering

Examples

Amazon works with 3 package delivery companies, and Fedex is giving a 10% discount for a year; should it use that?

- Easy business decision

Amazon changes its rec. algorithm to prefer geographically nearby sellers for each item, and profits decrease

- Is the problem incorrect data about seller locations?
- How are we defining geographically near: is it in miles, ship time, etc?
- Did something else cause profits to fall?
- Have sellers or buyers adapted to the change in policy?

Common Challenges in Using Data

Errors at source

Errors loading some of the data, or moving between systems

Inconsistent definition / schema in different locations

Inconsistent downstream calculation

Wrong calculation, metric, etc

Security, governance, performance, cost

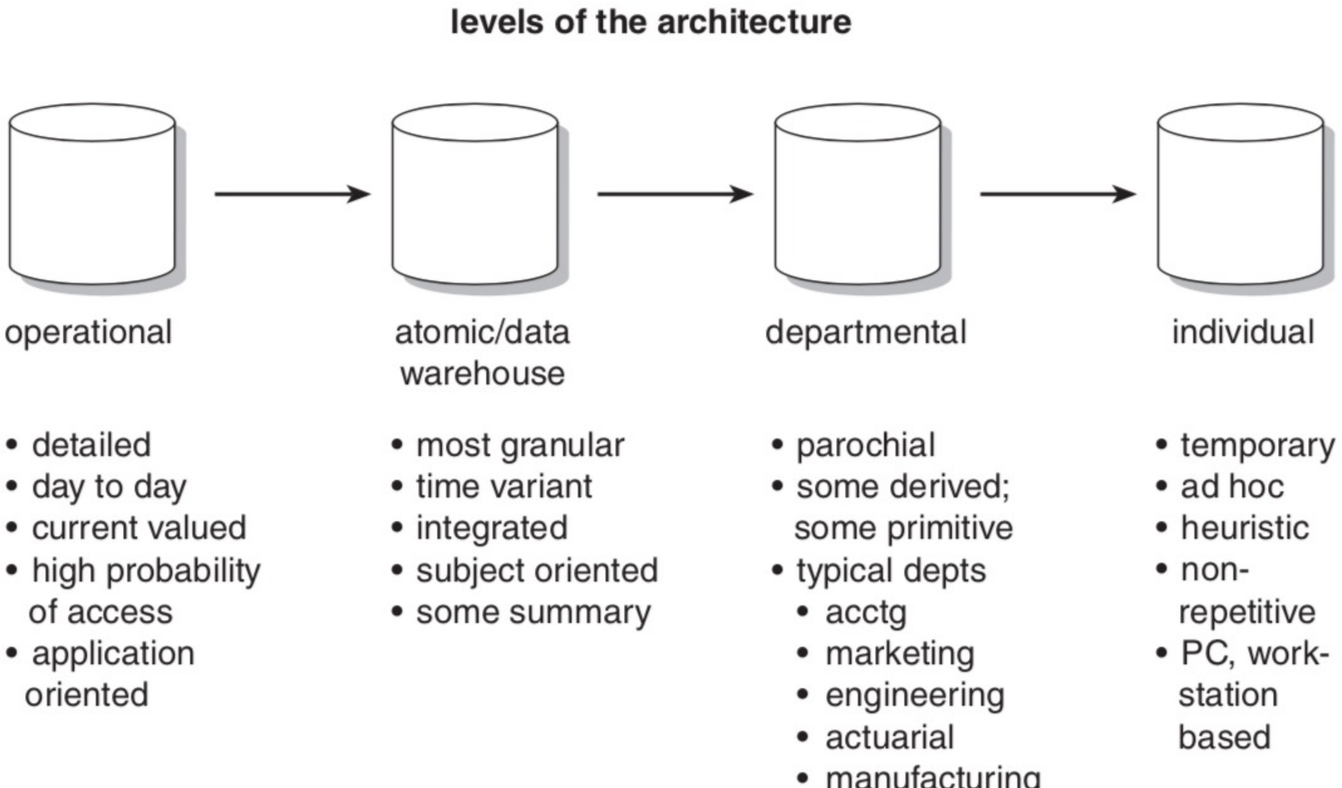
How can we address these challenges?

Idea 1: Data Architectures for Analytics

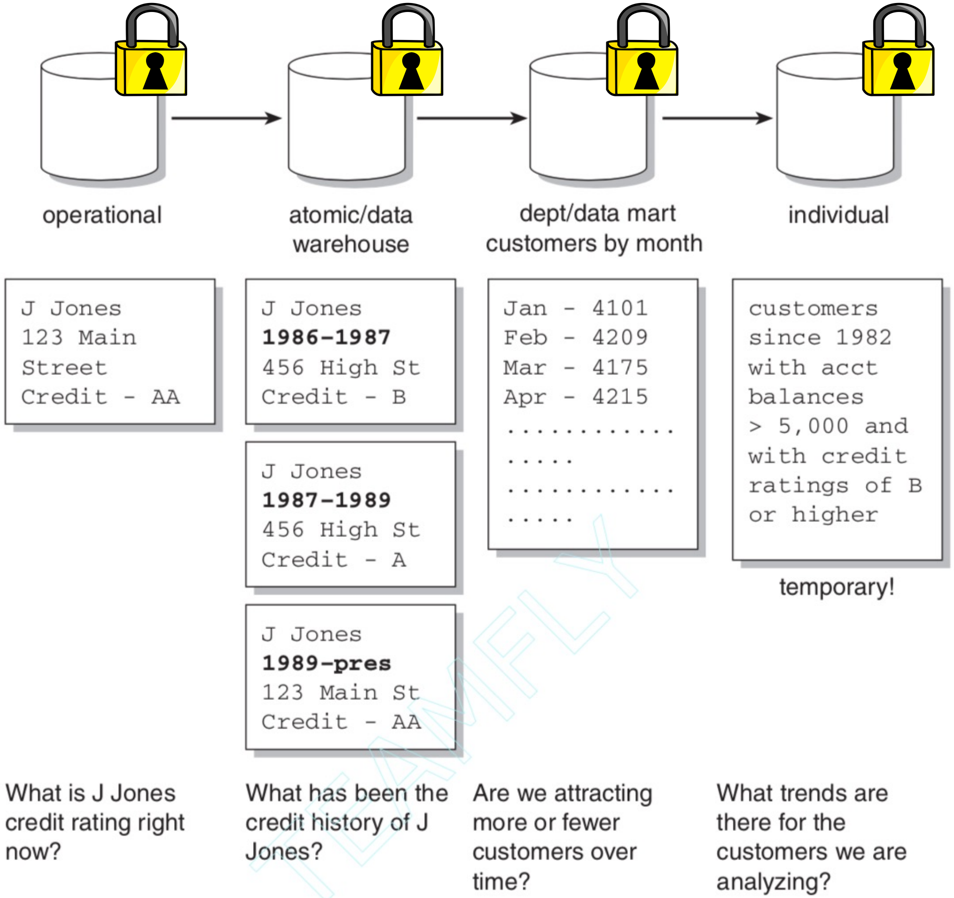
(Data Warehouses, etc)

PRIMITIVE DATA/OPERATIONAL DATA	DERIVED DATA/DSS DATA
• application oriented	• subject oriented
• detailed	• summarized, otherwise refined
• accurate, as of the moment of access	• represents values over time, snapshots
• serves the clerical community	• serves the managerial community
• can be updated	• is not updated
• run repetitively	• run heuristically
• requirements for processing understood a priori	• requirements for processing not understood *a priori*
• compatible with the SDLC	• completely different life cycle
• performance sensitive	• performance relaxed
• accessed a unit at a time	• accessed a set at a time
• transaction driven	• analysis driven
• control of update a major concern in terms of ownership	• control of update no issue
• high availability	• relaxed availability
• managed in its entirety	• managed by subsets
• nonredundancy	• redundancy is a fact of life
• static structure; variable contents	• flexible structure
• small amount of data used in a process	• large amount of data used in a process
• supports day-to-day operations	• supports managerial needs
• high probability of access	• low, modest probability of access

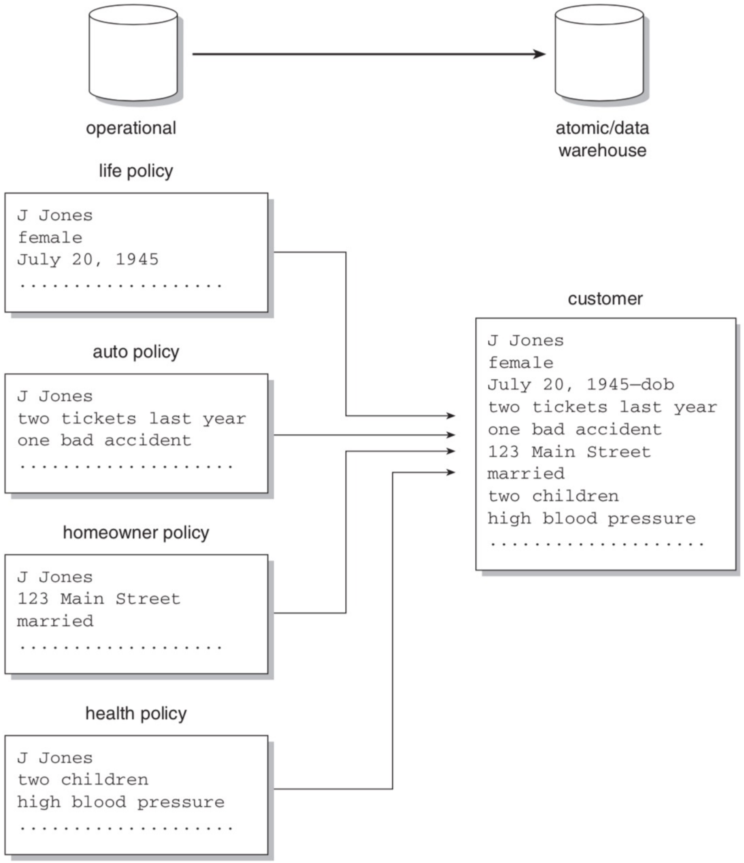
Idea 1: Data Architectures for Analytics



Idea 1: Data Architectures for Analytics



Idea 1: Data Architectures for Analytics



Idea 1: Data Architectures for Analytics

Many implementations today:

- Data warehouse systems (Teradata, Amazon Redshift, BigQuery, …)
- Data lakes for raw storage (Amazon S3, Apache Hadoop, …)
- Integrated products for specific use cases (Salesforce, SAP, …)

Data warehouses

Data lakes

Integrated apps

Other analytics apps

Idea 2: Data Engineering

Apply software engineering principles to data pipelines & outputs

- Testing (at development time)
- Monitoring (at runtime)
- Type checking (schemas, key constraints, etc)
- Separate dev, staging and production environments
- Upgrades with support for rollback, data versioning
- Data pipelines as code

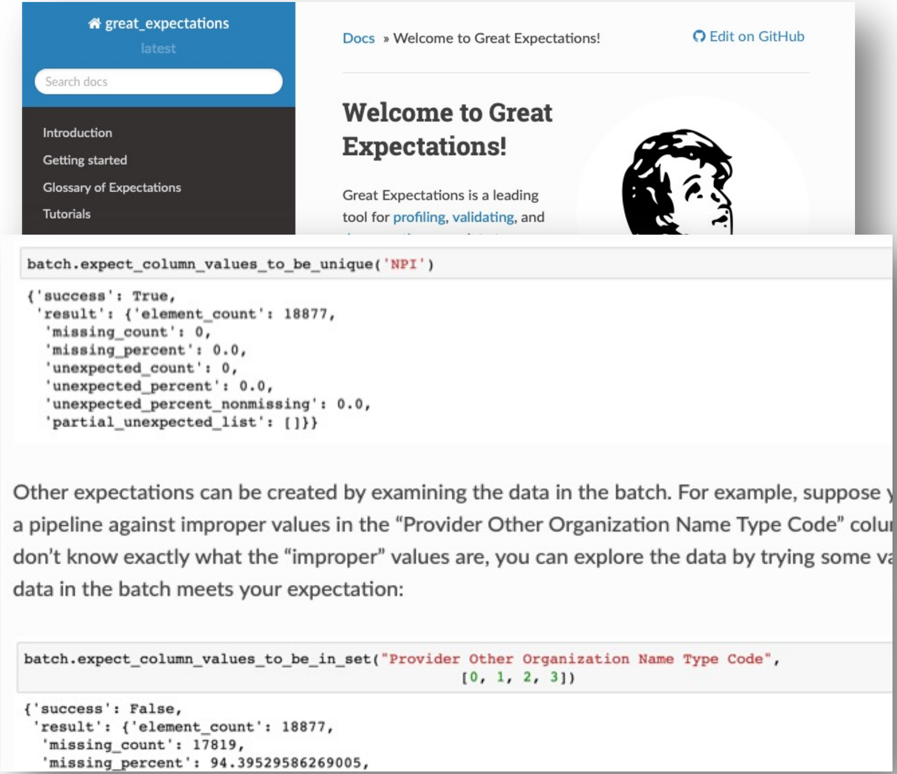
Idea 2: Data Engineering

Historical: database admins and analysts working with structured data in in SQL

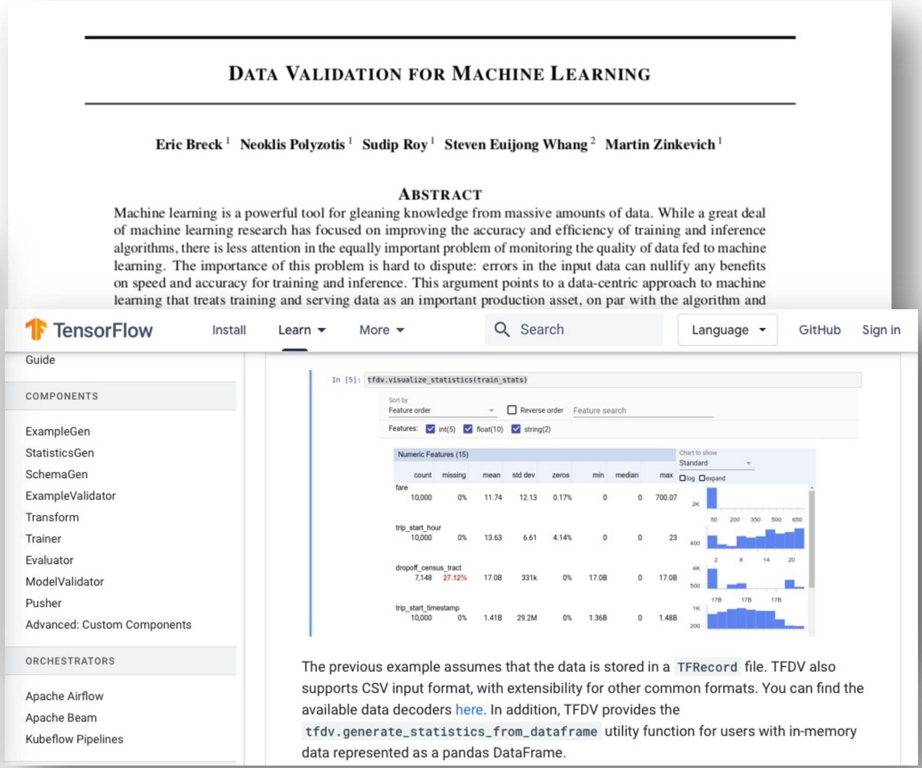
Today: expanded to more data types and more code, in programming languages such as Python & Java plus SQL

- The term “data engineer” is fairly new

Example Data Engineering Tools/Ideas



[Great Expectations]



[TFX Data Validation]

Outline

History of information systems

Challenges in using data

Example solutions

New trends & Databricks case study

What's Changing in Data & ML?

Rise of unstructured and "big" data

Originated in web companies but now used well beyond them

- Unstructured data means not tables: for example, images, video, audio, text documents, etc → often the largest data by volume
- New source of "big" tabular data: machine-generated data (e.g. timeseries)

New way to deliver computing technology: public cloud

Rent computer hardware and software as a service (SaaS), pay by usage (e.g. by hour)

Advances in data science & ML

Often enabled by data

Databricks

Founded 2013 by Berkeley researchers that created Apache Spark (open source big data computing engine)

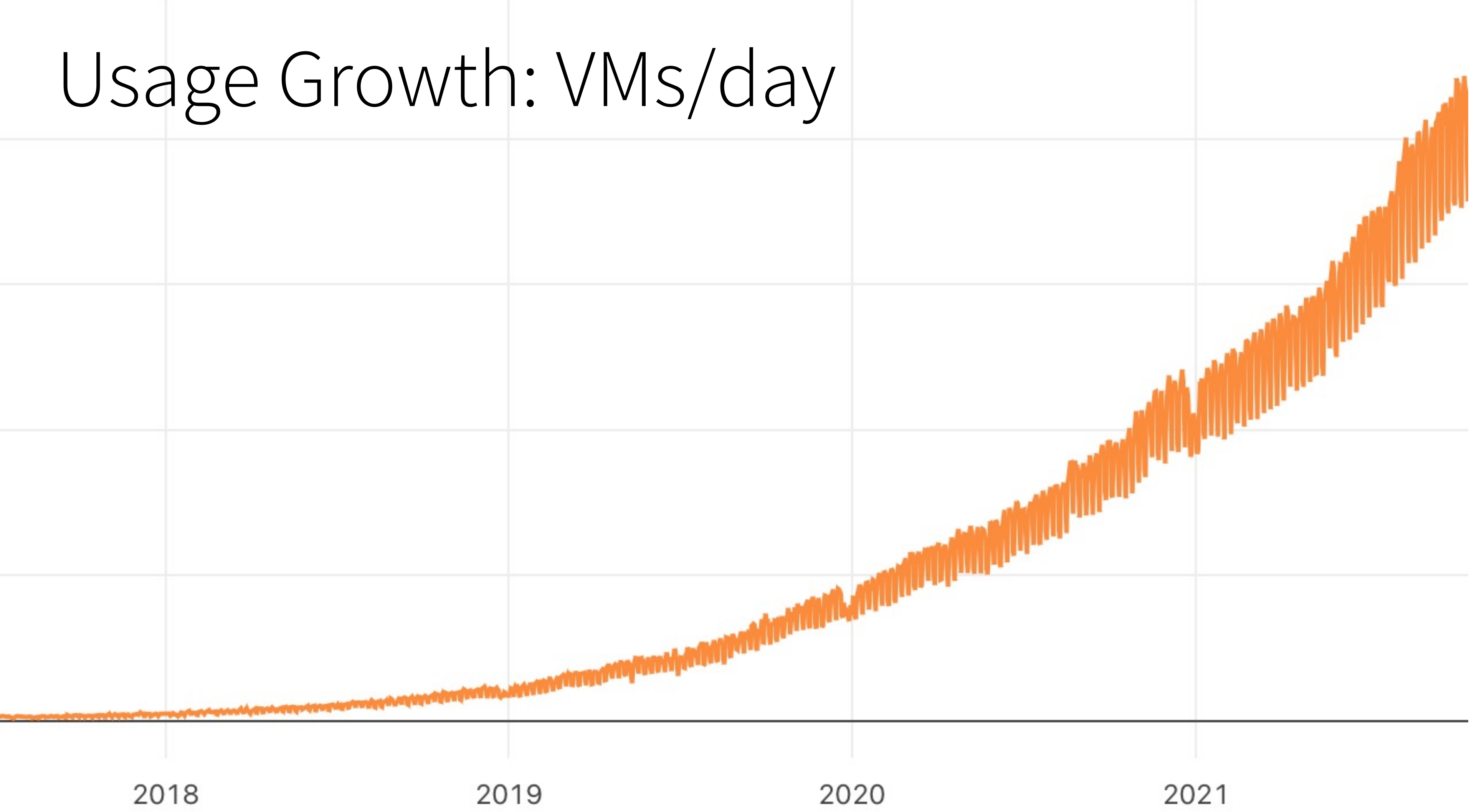
Business model: cloud service over AWS, Azure, etc

- Run & manage computing workloads for customers

>5000 customers, ~2000 employees

- 10+ million VMs processing exabytes of data per day

Usage Growth: VMs/day



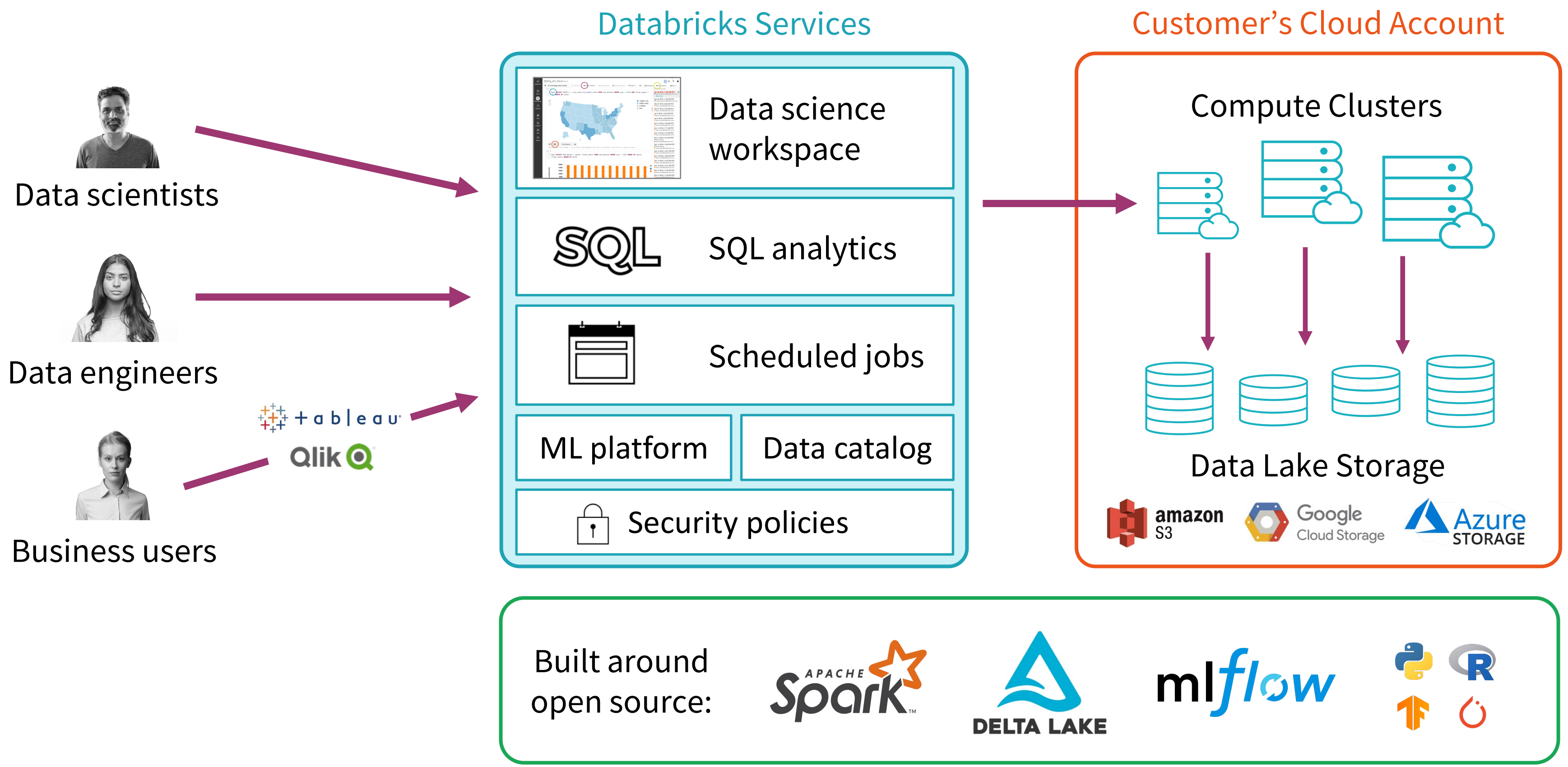
Example Use Cases

REGENERON — Correlate 500,000 patient records with DNA to design therapies

Shell — Optimize inventory management using simulations and ML

FINRA — Identify securities fraud via machine learning on 30 PB of data

Databricks Product: "Lakehouse" Platform



Takeaways from Databricks

1. Simplify building production data apps
2. Rise of cloud computing
3. Interesting use cases and challenges

Takeaways from Databricks

1. Simplify building production data apps
2. Rise of cloud computing
3. Interesting use cases and challenges

Simplify Building Production Data Apps

Most data apps have yet to be written, especially for new data sources (e.g. big data) or new techniques (e.g. ML)

Most of the effort in these apps goes to reliably combining and preparing the relevant data (in a way that keeps running later!)

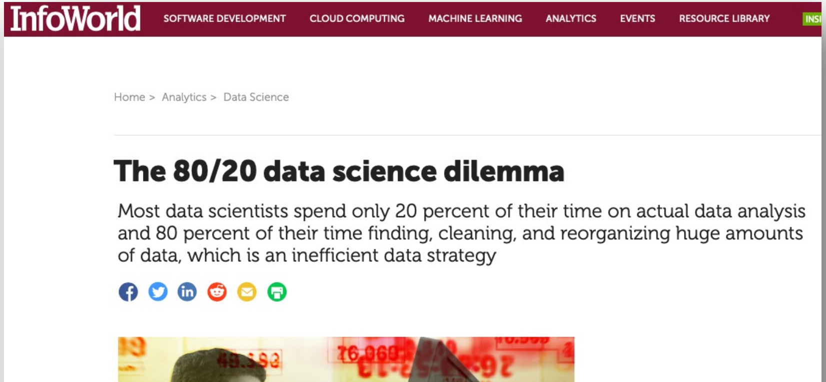
Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says

Gil Press Contributor
Big Data
I write about technology, entrepreneurs and innovation.

This article is more than 2 years old.

TWEET THIS

data scientists found that they spend most of their time massaging rather



InfoWorld SOFTWARE DEVELOPMENT CLOUD COMPUTING MACHINE LEARNING ANALYTICS EVENTS RESOURCE LIBRARY

Home > Analytics > Data Science

The 80/20 data science dilemma

Most data scientists spend only 20 percent of their time on actual data analysis and 80 percent of their time finding, cleaning, and reorganizing huge amounts of data, which is an inefficient data strategy

Simplify Building Production Data Apps

Most data apps have yet to be written, especially for new data sources (e.g. big data) or new techniques (e.g. ML)

Most of the effort in these apps goes to reliably combining and preparing the relevant data (in a way that keeps running later!)

→ Figure out ways to let **more people** build **reliable** apps

Example: Data Science Report

Alex the data scientist wants to publish a sales forecast each day based on machine-generated & customer data

Before Databricks: Alex needs to work with a data engineer and DB admin to create data loading jobs in Java and SQL, then get them to deploy and run these jobs

After Databricks: Alex can use Spark API to access both unstructured & structured data in a Python notebook, schedule the notebook as a job, and run it on her own cloud VMs

Big difference in productivity even at eng-heavy companies

Example: Business Intelligence

Before Databricks: Business analysts need to wait for data to be loaded into a data warehouse to do analytics in Tableau (and even then, only a subset of data is in there)

After Databricks: Business analysts can connect Tableau to the entire data lake and query all data in the organization as soon as it arrives

Focus on maximizing **access**, **reliability** & **timeliness** of data

Production Gap in Machine Learning

ML Research & Courses

Goal: designing a good model

Data is provided and ready to use
(e.g. benchmark dataset)

No need to deploy, monitor, retrain or implement governance

Tools for model design & evaluation
(e.g. TensorFlow, SciKit)

ML Products

Goal: reliably solving a business problem

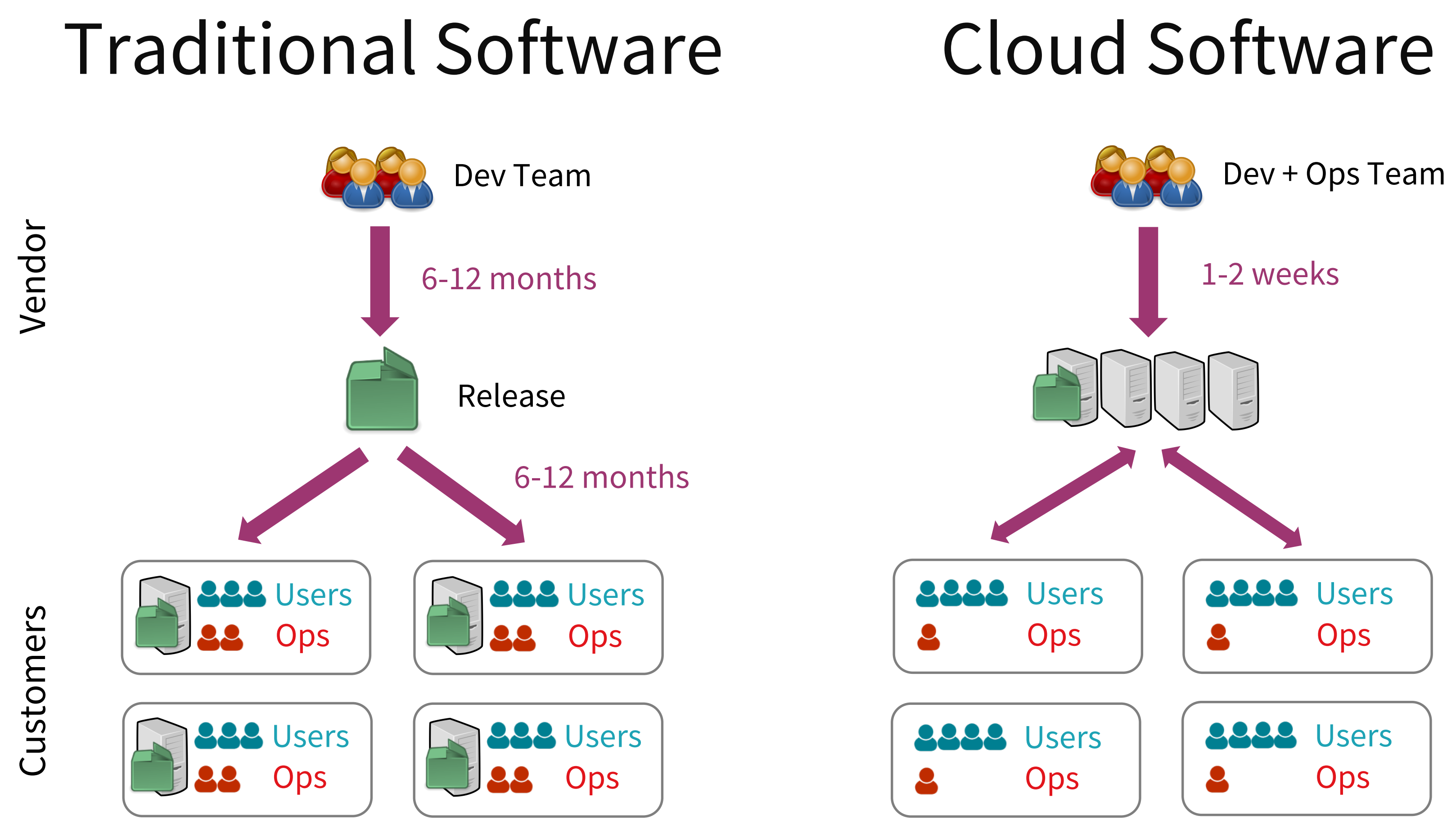
Data is often the biggest challenge
(for models, try many standard ones)

Must continuously deploy + monitor + retrain, and have governance

Need new tools to enable this process!
(ML platforms such as MLflow)

Takeaways from Databricks

1. Simplify building production data apps
2. Rise of cloud computing
3. Interesting use cases and challenges


Traditional Software
Cloud Software
Vendor
Customers
Dev Team
6-12 months
Release
6-12 months
Users
Ops
Users
Ops
Users
Ops
Users
Ops
Dev + Ops Team
1-2 weeks
Users
Ops
Users
Ops
Users
Ops
Users
Ops

Why Use Cloud Software?

1. **Management built-in:** much more value than the software bits alone (security, availability, etc)

2. **Elasticity:** pay-as-you-go, scale on demand

3. **Better features** released faster

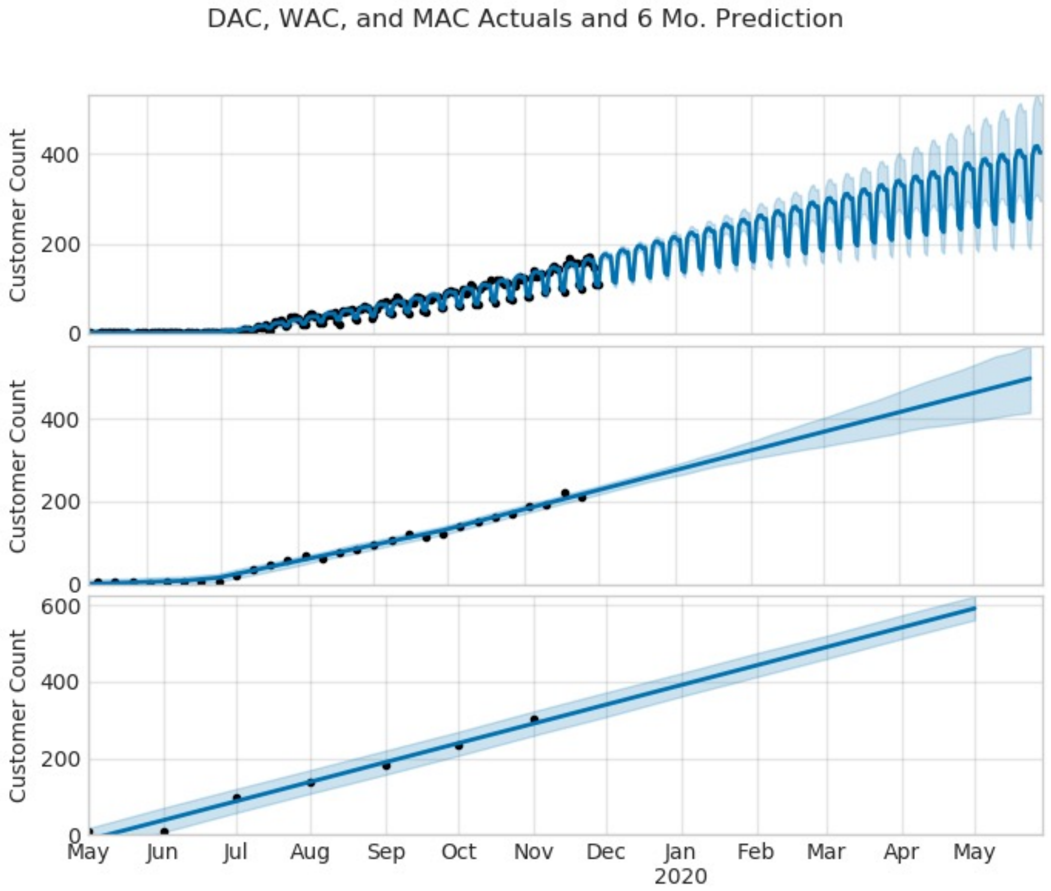
Differences in Building Cloud Software

+ Rapid feedback: release updates any time, monitor usage live
+ Only need to maintain 2 software versions (current & next), in fewer configurations than you'd have on-premise
- Updating without regressions: very hard, but critical in cloud (includes API, semantics, and performance regressions)
- Operating a multitenant service: scaling, security, user isolation

Example Uses of Telemetry at Databricks

"Data McNuggets": automated report for 20+ product features

Adoption Predictions for Warm Pools



Example Uses of Telemetry at Databricks

Top 10 Customers

This section covers the top 10 customers that are using this product.

customer	cse	Number_of_Pools (in 10s)
[redacted]	[redacted]	42
[redacted]	[redacted]	41
[redacted]	[redacted]	37
[redacted]	[redacted]	35
[redacted]	[redacted]	27
[redacted]	[redacted]	26
[redacted]	[redacted]	26
[redacted]	[redacted]	25
[redacted]	[redacted]	24
[redacted]	[redacted]	22

Customers who stopped using Warm Pools.

This section covers the list of 10 recent customers who have stopped using this product for 7 days and their last date using this product.

end_date	customer
2019-11-25	[redacted]
2019-11-25	[redacted]
2019-11-25	[redacted]
2019-11-24	[redacted]
2019-11-22	[redacted]
2019-11-22	[redacted]
2019-11-22	[redacted]
2019-11-22	[redacted]
2019-11-22	[redacted]
2019-11-22	[redacted]

Takeaways from Databricks

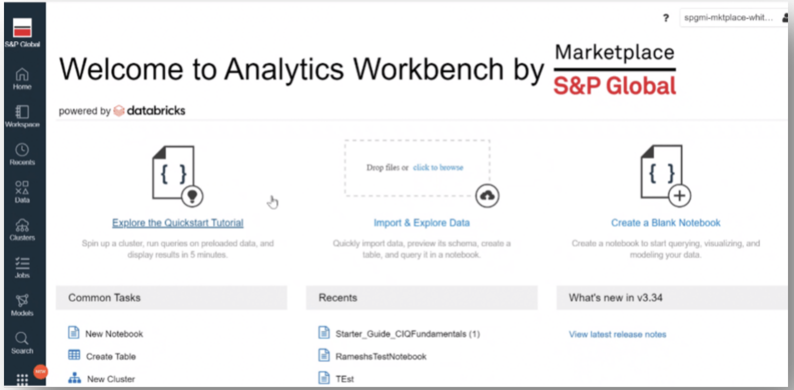
1. Simplify building production data apps
2. Rise of cloud computing
3. Interesting use cases and challenges

Interesting Databricks Use Cases

Selling data products to other companies

- **Traditional approach:** charge for monthly access to new data
  - Little room for price discrimination: everyone gets full datasets
- **New approach:** give users a hosted computing platform and charge by query, by seats, or by other metrics (OEM Databricks!)

Example: S&P Marketplace Workbench



Interesting Databricks Use Cases

Surprising sources of big data

- **Network security:** log every network operation to catch intruders, malicious insiders, exfiltration, etc → petabytes of data per day
- **Public datasets:** many biotech and financial companies use these
- **Industrial IoT:** every factory device, airplane engine, etc is instrumented

New Trends and Challenges

Privacy regulations, such as GDPR and CCPA, have required companies to significantly rethink data systems (for the better?)

- Only collect & use data for specific purposes (how to track this?)
- Any user can ask to delete their data (can’t store it immutably as before!)
- B2B contracts also increasingly have complex rules

New Trends and Challenges

Production machine learning: ML apps have similar problems to data applications (too much data for human review), but worse:

- Opaque and misaligned metrics
- Overfitting
- Bias

Conclusion

Using data in products and business decisions is hard: need to define, collect, and monitor the data reliably and analyze in depth

Various technologies and principles for this have emerged over time (data engineering, data warehouse architecture, etc)

Lots of room to make this simpler