Peregrine: workload optimization for cloud query engines

Alekh Jindal, Hiren Patel, Abhishek Roy, Shi Qiao, Zhicheng Yin, Rathijit Sen, Subru Krishnan

Microsoft


Engine
Workload
DBA


On-Premise
DBA


On-Premise
DBA


On-Premise
Need to reach by 10,
can we drive faster?
Sure!
TAXI
ТАКСИ
DBA

Cloud Query Engines

- Setup, installation, maintenance taken care of
- On-demand provisioning, pay as you go

Reality Check for providers:

- System developers == virtual DBAs!
- Too many cloud users, compared to system developers
- Too many support requests; often redundant
- Less time for feature development

Reality Check for customers:

- Lots of services to choose from (even within Azure, GCP, AWS)
- Lot of knobs to tune for **good perf** and **low cost**
- Lack of control; and lack of expertise
- And, **<u>the DBA is gone</u>**!

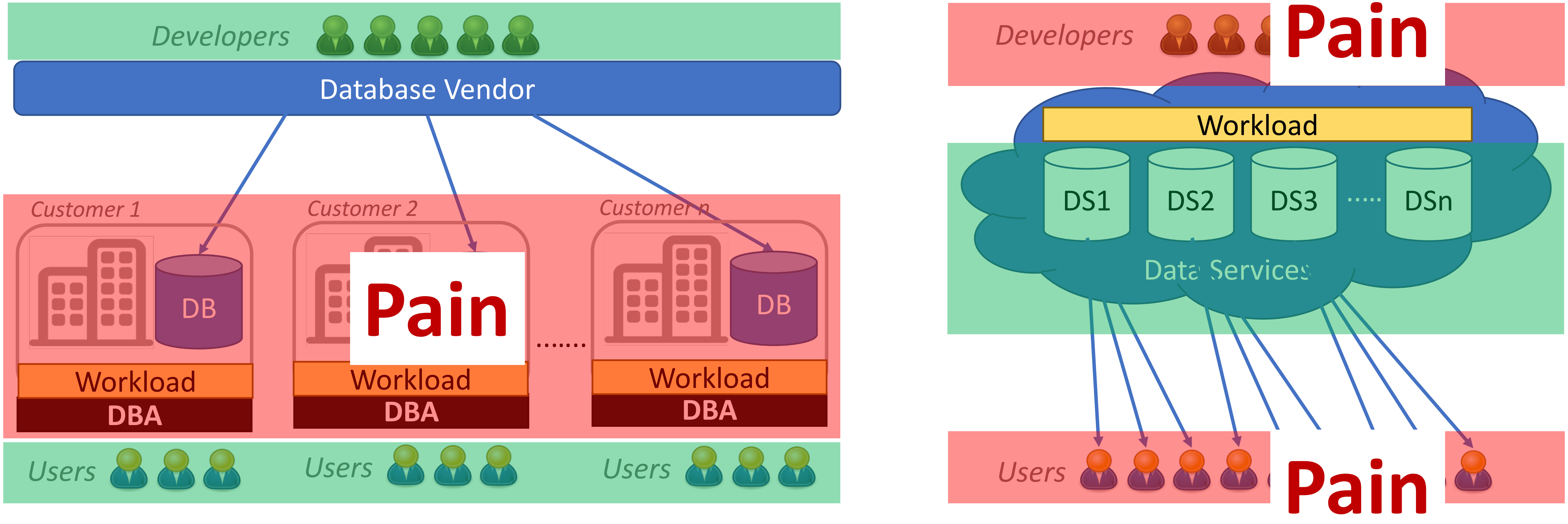
Cosmos: big data infra at Microsoft

- 100s of thousands of machines
- Exabytes of data at rest; Petabytes ingress/egress daily
- 500k+ batch jobs / day
- 3B+ tasks executed / day
- 10s of millions interactive queries / day
- 10s of thousands of SCOPE developers
- 1000s of teams

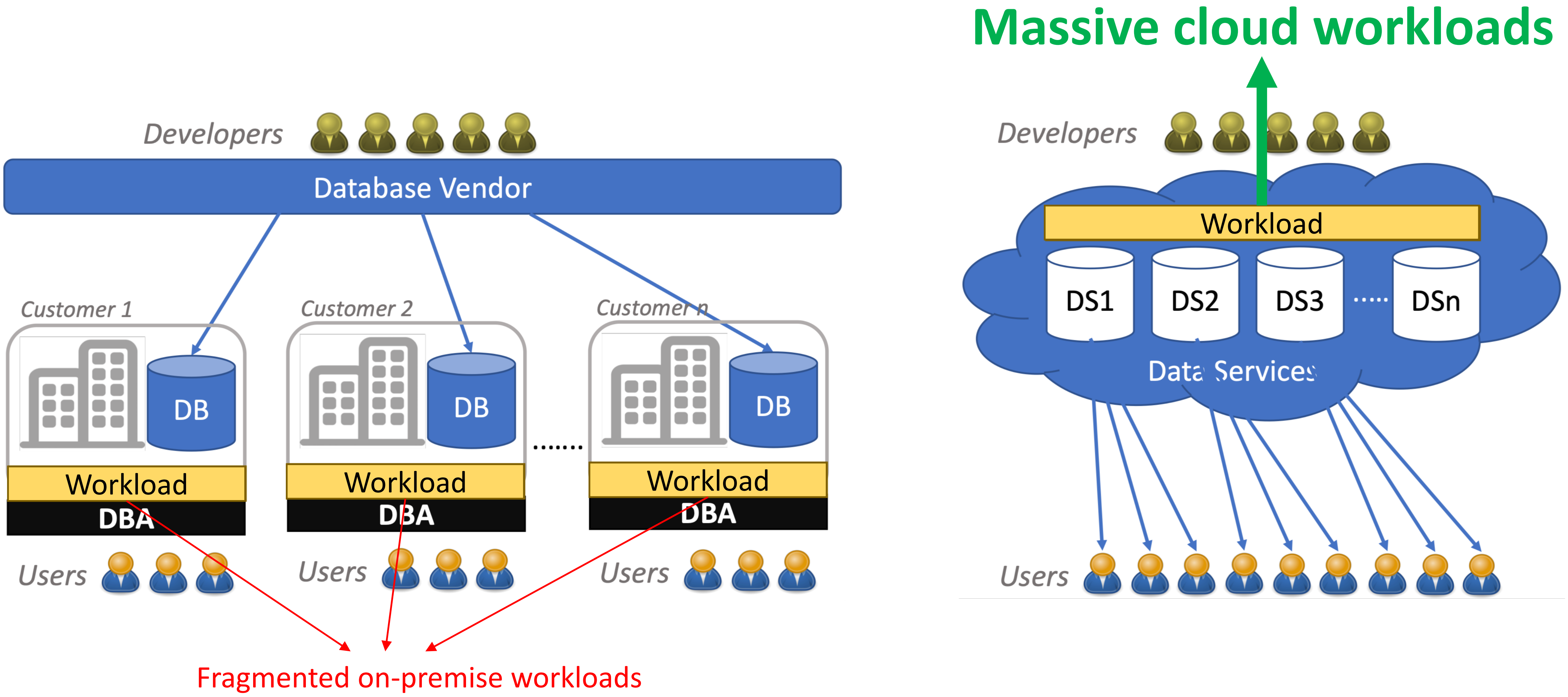
The missing DBA and the growing pain in Cosmos

- Large number of knobs/hints at script, data, plan level
  - Only few expert users
  - Rest need guidance
  - Survey: better tooling for improving SCOPE queries
- Support challenge
  - 10s of thousands incidents / years
  - 10 incidents per system developer on call
  - 100x users compared to system developers
  - ~10% growth in SCOPE workload in 2019

The cloud pain



The cloud opportunity



The Cosmos opportunity

Job metadata
name, user, account, submit/start/end times

Query plans
logical, physical, stage graph, estimates

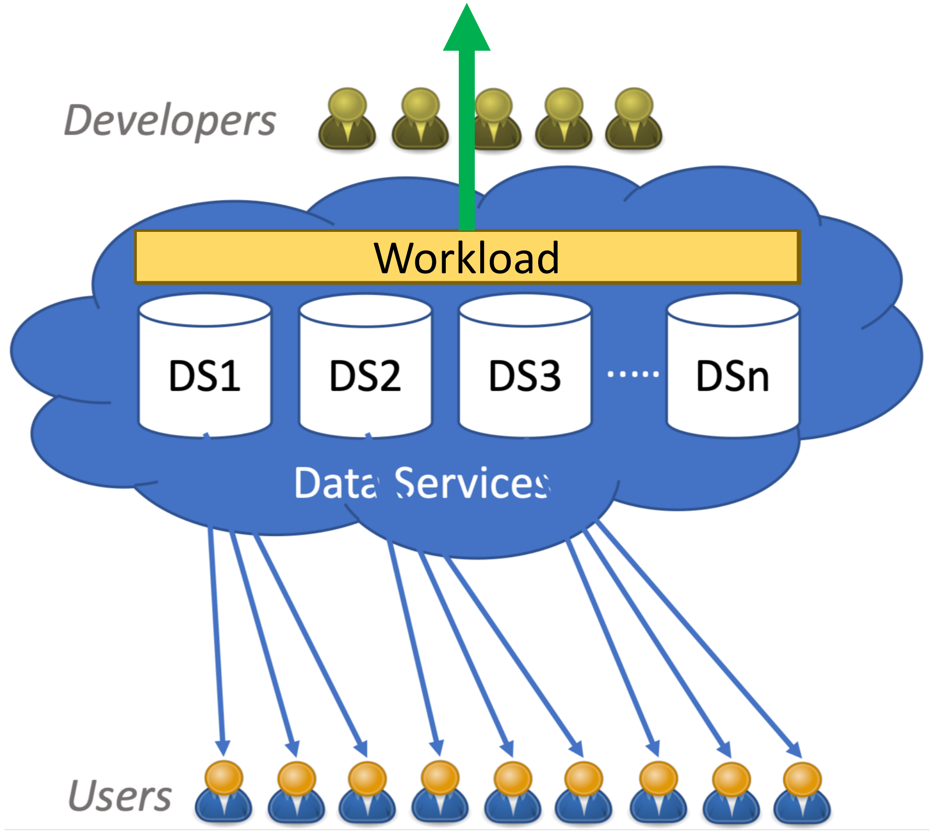
Runtime statistics
Operator-wise observables

Task level logs
start/end events

Machine counters
CPU, IO, etc.

Several TBs of metadata / day

Massive cloud workloads



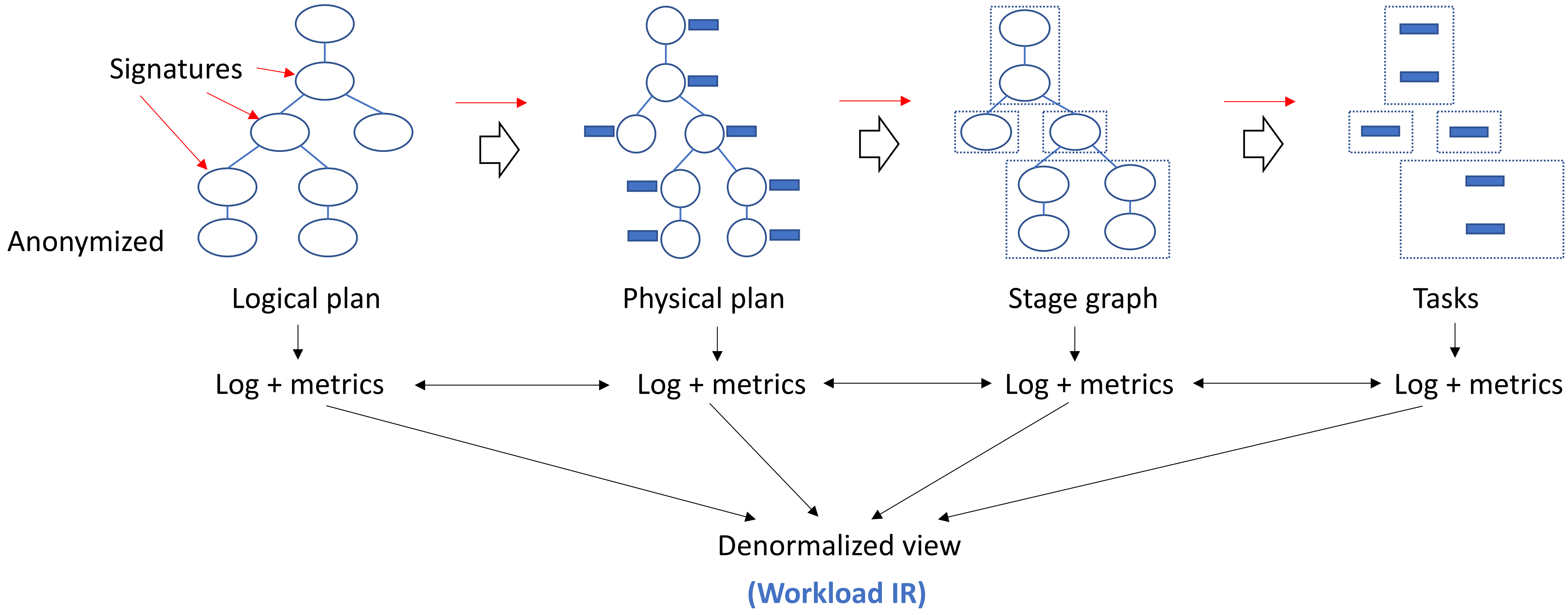
The case for a workload optimization platform

- DBA-as-a-Service
  - Another service in the cloud (easier integration)
  - Based on cloud workloads at hand (instance optimization)
- Engine agnostic
  - Not specific to different query engines, e.g., SCOPE, Spark, SQL DW, or etc.
  - E.g., view selection is still the same problem
- Global optimizations
  - Cloud workloads are organized into data pipelines
  - People often care about end-to-end aggregate costs in the cloud

Step 1: workload representation

Instrument, log, and collect workload characteristics

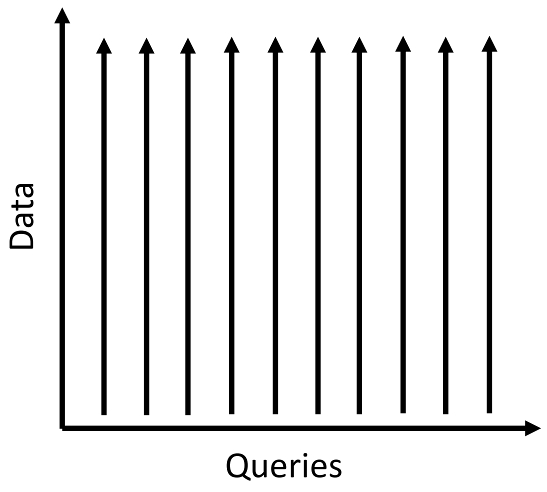
Engine-agnostic workload representation



Step 2: optimize for patterns

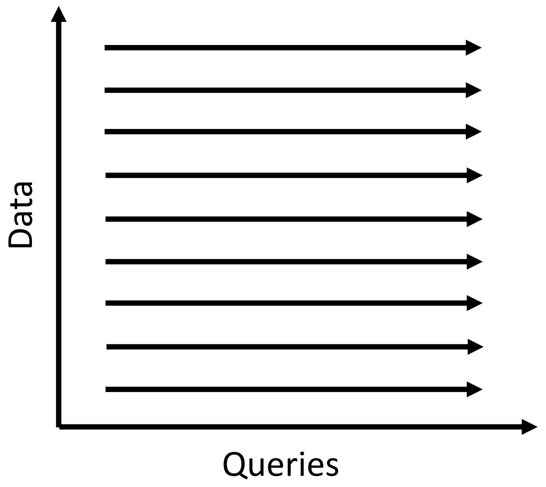
Typical workload patterns

- Consider a simplified 2D space of data and queries



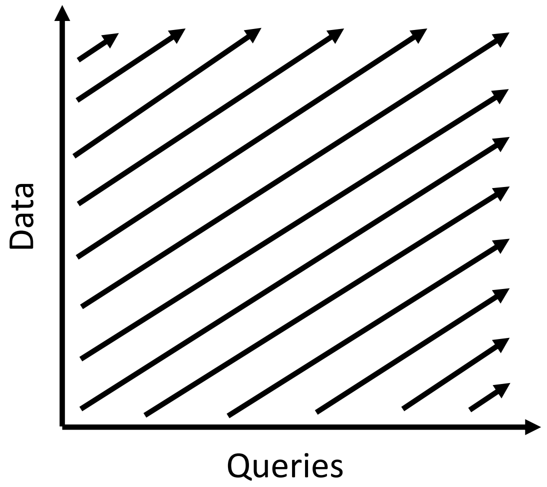
Recurring

Query templates appear over newer datasets



Similarity

Queries over same datasets have similarities

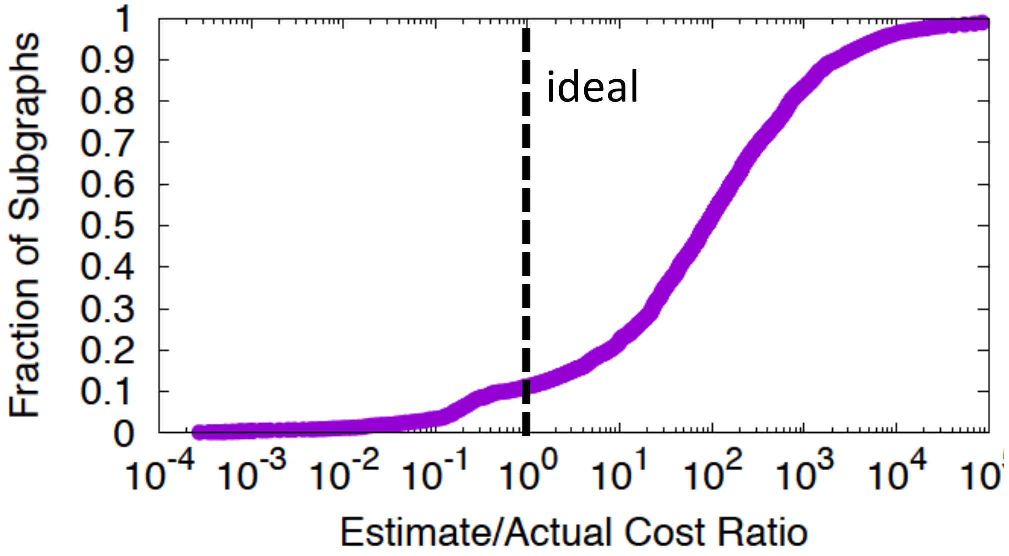


Dependency

Queries depend on datasets produced by previous queries

Recurring pattern

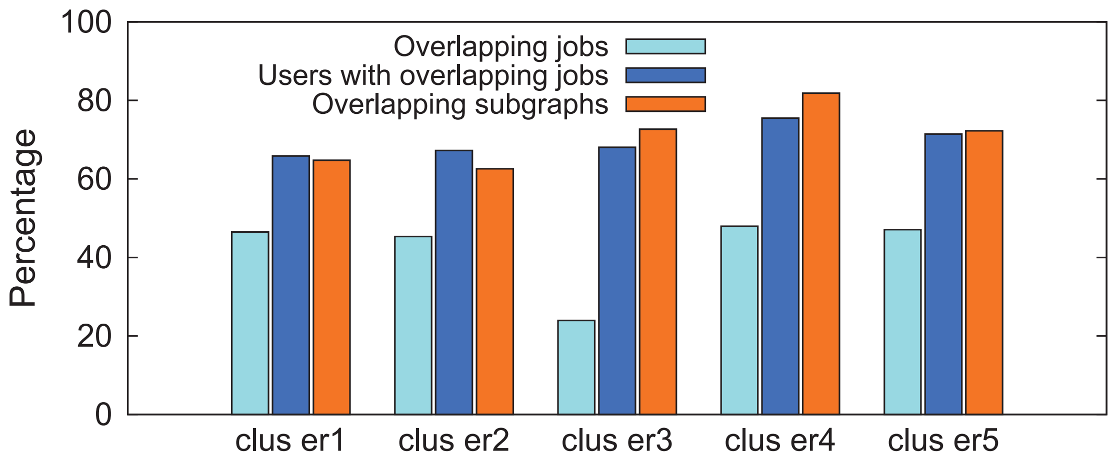
- Majority of production workloads
  - There is a regular ETL needed before other things can happen
- Opportunity to learn from the past
- Examples
  - Learned cardinality*
  - Learned cost models
  - Learned resources
  - Learned etc.



* **Towards a Learning Optimizer for Shared Clouds**. Chenggang Wu, Alekh Jindal, Saeed Amizadeh, Hiren Patel, Wangchao Le, Shi Qiao, Sriram Rao. *VLDB 2019*.

Similarity pattern

- Very typical in multi-user shared cloud environments
  - Cosmos, HDI, Ant Financial, ML workflows, etc.
- Opportunity for multi-query optimization
- Examples
  - CloudViews*
  - Checkpointing
  - Caching
  - Etc.



* **Computation Reuse in Analytics Job Service at Microsoft**. Alekh Jindal, Shi Qiao, Hiren Patel, Jarod Yin, Jieming Di, Malay Bag, Marc Friedman, Yifung Lin, Konstantinos Karanasos, Sriram Rao. *SIGMOD 2018*.

* **Selecting Subexpressions to Materialize at Datacenter Scale**. Alekh Jindal, Konstantinos Karanasos, Sriram Rao, Hiren Patel. *VLDB 2018*.

Dependency pattern

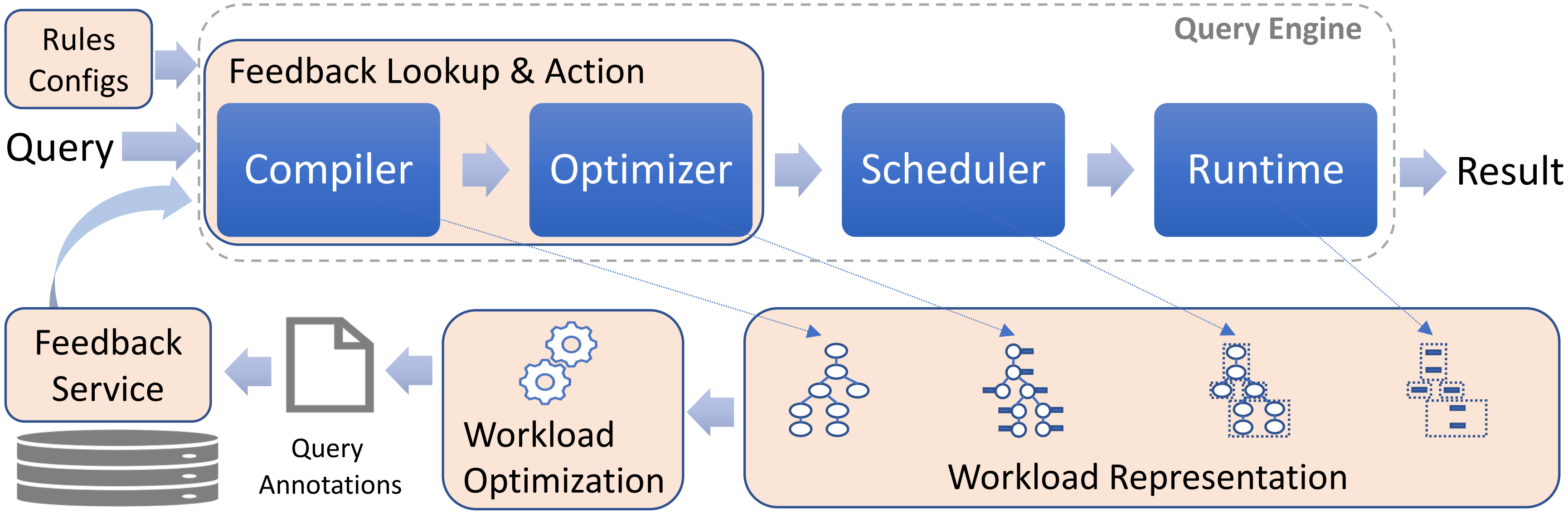
- Queries are typically organized in pipelines
  - Smaller steps that are easier to build and maintain
- Dependency driven optimizations/analytics*
  - Relative importance of jobs for scheduling
  - Physical design tuning
  - Etc.

* **Dependency-driven analytics: A compass for uncharted data oceans**. R. Mavlyutov, C. Curino, B. Asipov, and P. Cudré-Mauroux. *CIDR 2017.*

Step 3: feeding it back

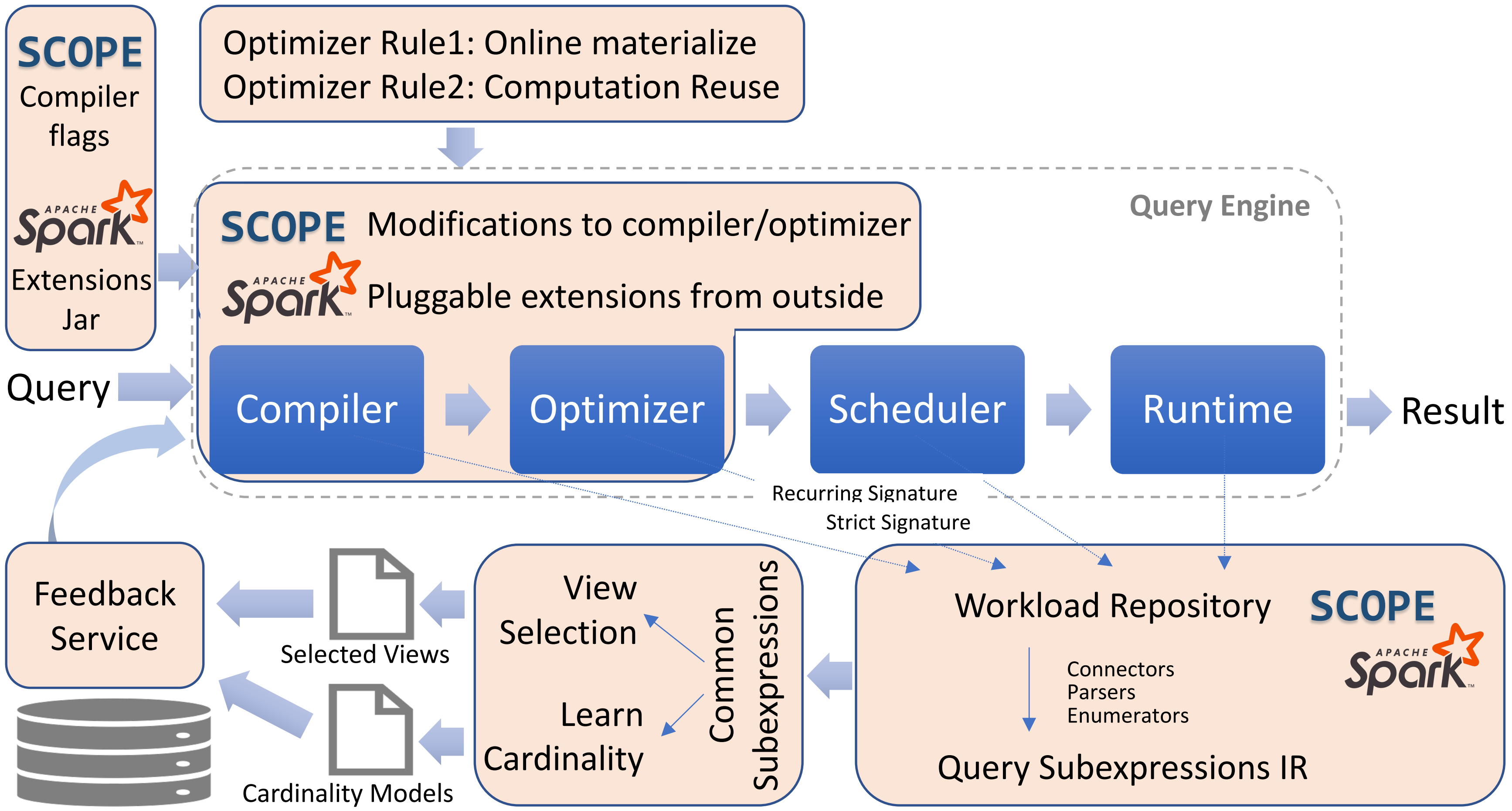
- Actions
  - Insights
  - Recommendations
  - Self-tuning

Self-tuning



Annotation: *signature --> actions*

Illustration: Scope and Spark query engines



The third axis: people

- Easier for people to play with the query workloads
  - Abstracts many of the painful steps
  - Allows people to build on top of each other
  - Focus more on the workload optimizations
- Enabled several
  - Researchers
  - Developers
  - Interns

Summary

- Gray Systems Labs (GSL)
  https://azuredata.microsoft.com/labs/gsl

- GSL@SoCC: 4 papers, 1 poster
- We are hiring!

Microsoft