Cloud Polystores Overview and Open Research Questions

Daniel Glake, Felix Kiehn, Mareike Schmidt

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

DBIS

Use Case I: E-Commerce

- Example: Otto Group, Amazon, …
- What do they need:
  - Availability
  - Consistency only for subset of entities
    - E.g. full consistency for completed orders
    - E.g. partial consistency for ratings and stock
  - High read throughput when querying for products
  - Complex Analytics (e.g. market basket analysis, …)

Use Case II: Simulation and decision support

- Example: Large scale traffic simulation, Game Engines, …
- What do they need:
  - High Write throughput (persisting results)
  - Realtime Data Visualization
  - Analytics with aggregated results
  - Support for standardized spatial formats

Use Case III: Medical Data Management

- Example: Hospital, Care facility, research centers, …
- What do they need:
  - High demand for consistency and data quality
  - Support for multimedia formats
  - Need for full-text search in previous diagnostics
  - Privacy and restricted access to data
  - Availability

Concluded Requirements I

- Multiple kinds of queries
  - Point and range queries
  - Filtering data
  - Full text search
  - …
- Special complex analytic queries
  - Reading and writing spatiotemporal data
  - Making time series analysis
  - Graph analysis
  - …

Concluded Requirements II

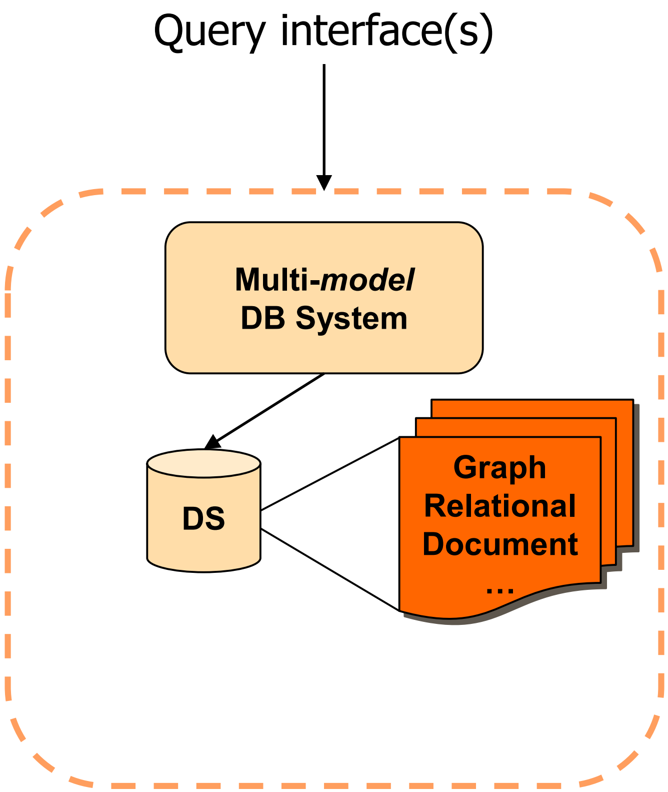
- Integrity constraints
- Ability to handle multiple data models
- Different degrees of consistency
- Different demands for availability of certain entities or subsets
- Different throughputs
  - High/Low reads
  - High/Low writes
- …

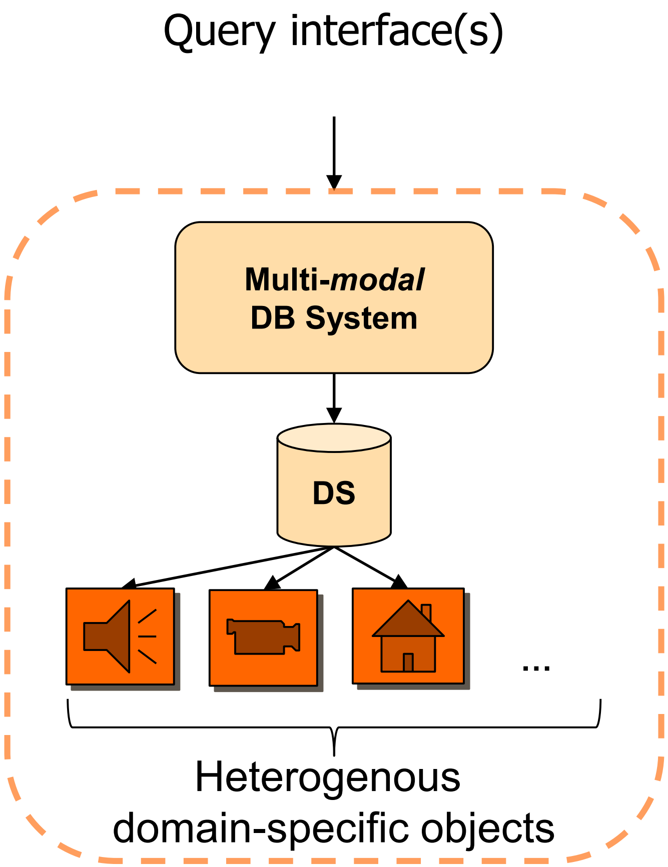
Cloud & Big Data Landscape

- Easy to get lost
- No "one size fits all"
- No standard
- Keeps evolving

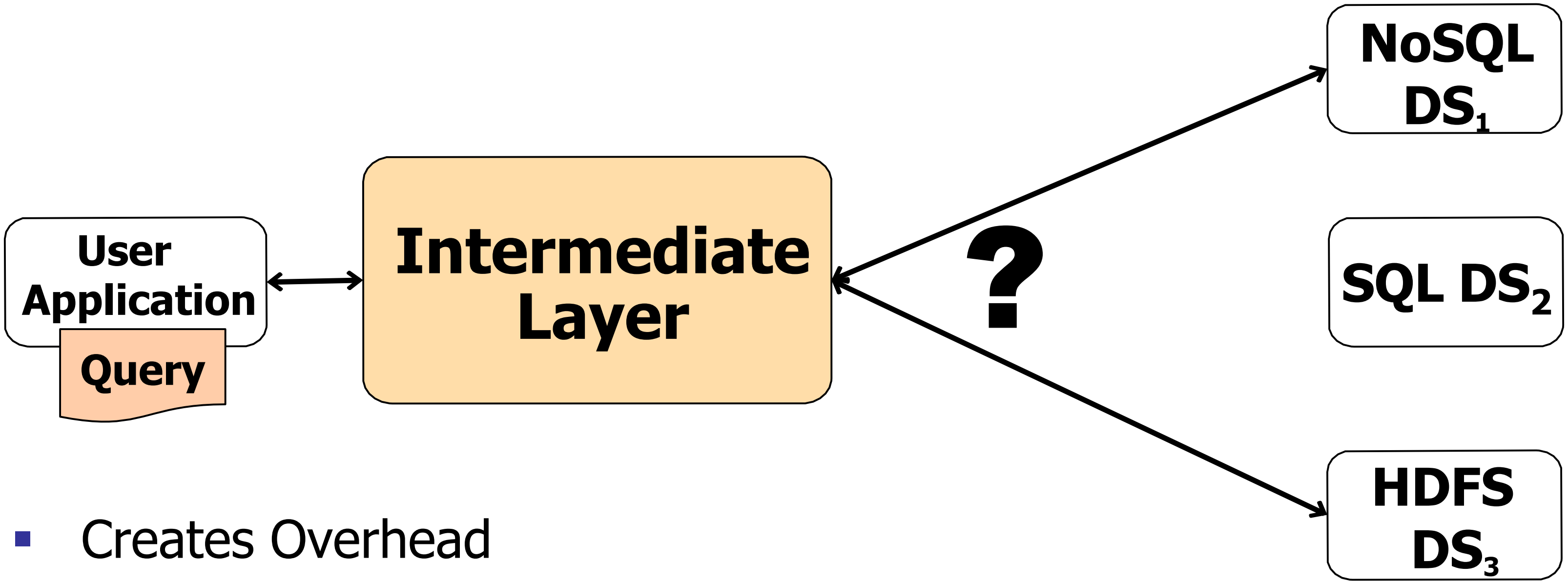
Source: https://www.forbes.com/sites/davefeinleib/2012/06/19/the-big-data-landscape/ (Retrieved: 4th June 2019)

Multi-model vs. Multi-modal



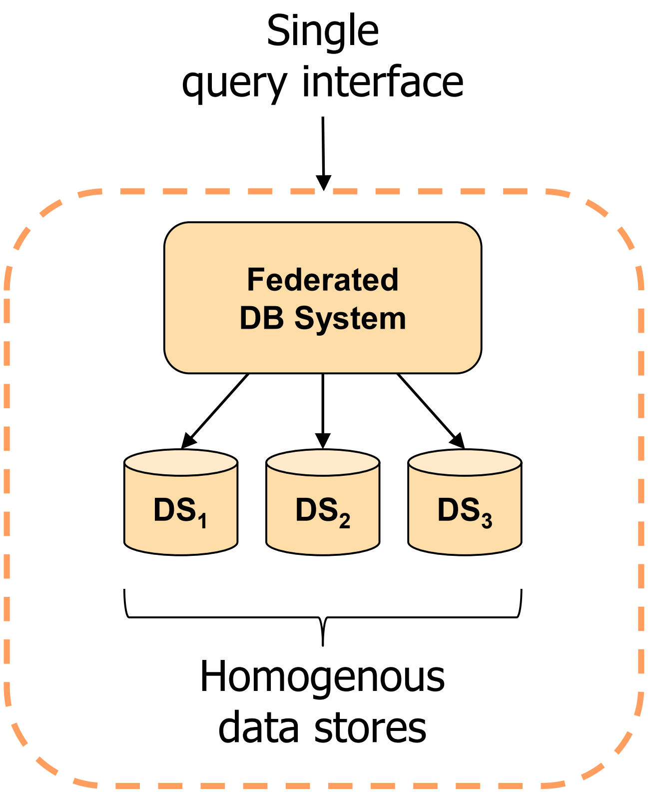


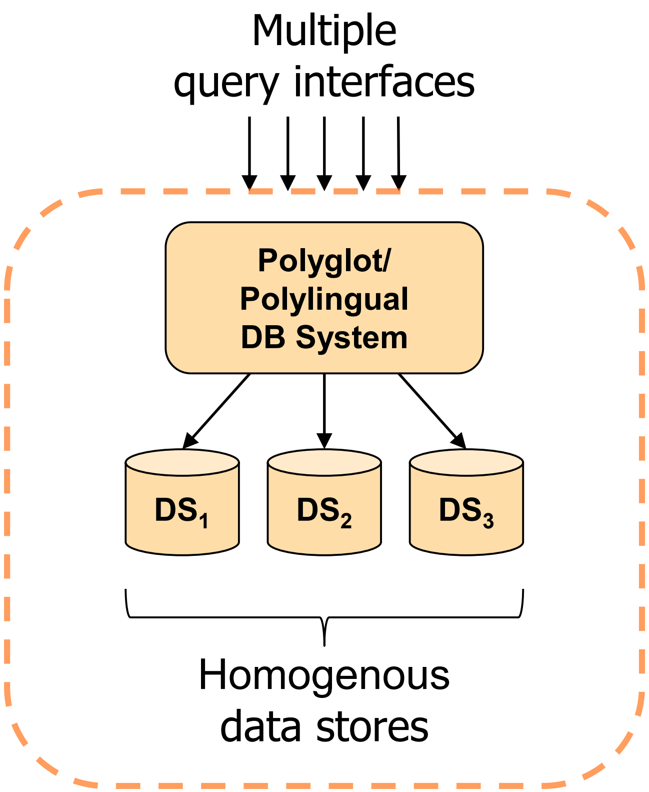
How to solve this with multiple DS?



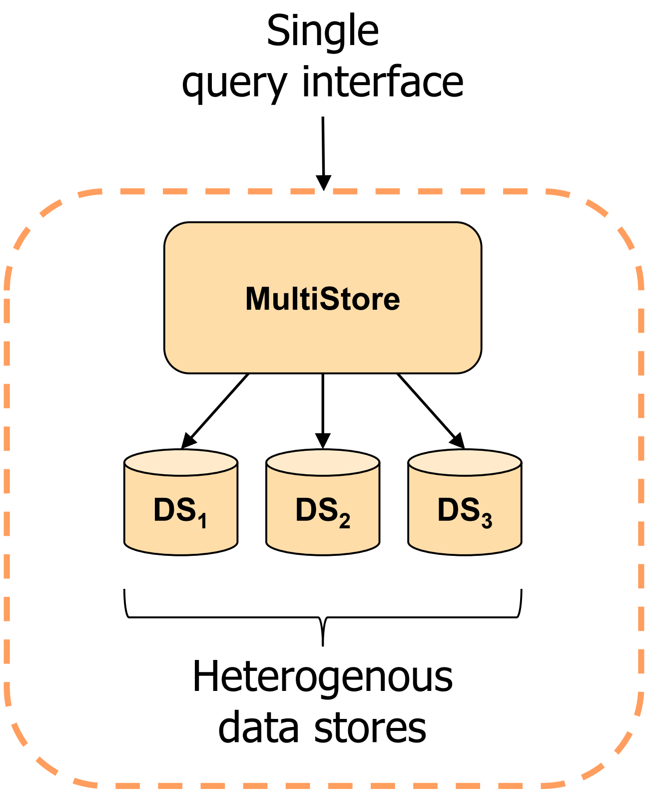
- Creates Overhead
  - Querying different databases
  - Managing intermediate results
  - Delivering (e.g. sorting) the final results
- Can performance be further enhanced despite the additional overhead?

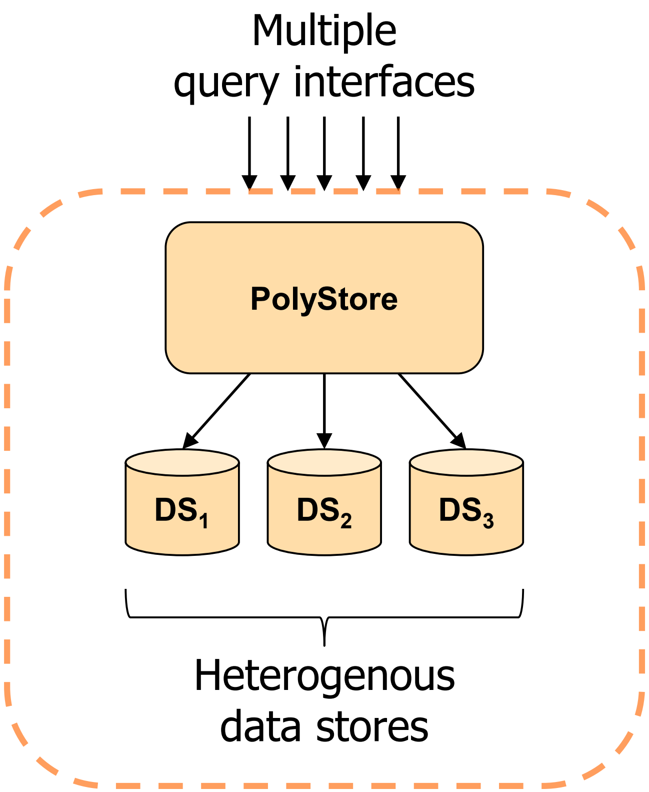
Federated system vs. Polyglot/-lingual system





MultiStore vs. PolyStore





Problems and Challenges

How to handle CAP/PACELC/BASE over different systems?

How t

How to do internal optimization and benchmarking?

How can replication and partitioning mechanisms be used?

How to realize extensibility?

How to handle schem

How to realize logging and recovery over mu

How to handle different semantics?

How to handle different consistency levels
within a multi-/polystore system?

the query planning a

Which data models should/can be supported by the system?

How to enable a system to be self-adaptive?

st-efficient data migration? (Data is slow!)

ity?

How to enable the user to define requirements for his/her data?

How to carry out distributed transaction over different systems?

How to provide full functionality of the underlying systems?

Is there a suitable common data model?
How to work with one?

How to enable data stores to communicate
to each other and work together?

How to enable the system to learn requirements on its own?

**Huge and diverse set of questions and challenges!
Thus, focus on
subset of problems.**

Explored Systems

	Design objective	System type	DI architecture
SQL++/ Forward	Unified relational and JSON-models	Query language and processor	GAV
CloudMdsQL	Query relational and NoSQL stores with Python extension	Query interface	Hybrid
Estocada	Extensible model integration	Multistore (Delegation system)	LAV
Polybase	Querying Hadoop Cluster over Microsoft-SQL RDBMS	Multistore (HDFS bridge)	LAV
BigDAWG	Unification of relational, NoSQL and NewSQL models	Polystore	Hybrid
Polypheny-DB	Self-Adapting Polystore	Polystore	

(Only) a Vision Paper

	SQL++/ Forward	Cloud MdsQL	Estocada	Polybase	BigDAWG	Polypheny-DB
Schema modeling (mult. DM)	✗	✗	✗	✗	(✓)	?
Query language	Single	Single	Single (QBT^{XM})	Single (T-SQL)	Multi (SQL, AFL, D4M with SQL)	Multi (SQL, Cypher, CRUD)
Write operations	✗	✗	✗	✓	✗	✓
Query optimization	Cost-based	Cost-based	Cost-based	Cost-based	Heuristic	Cost-based
Query execution	Query splitting	Bind join	View-based rewriting	Query splitting	Query splitting + Data shipping	Query splitting + data shipping
Semantic	Manual	Hybrid	Fixed	Fixed	Fixed	?

	SQL++/ Forward	Cloud MdsQL	Estocada	Polybase	BigDAWG	Polypheny-DB
Monitoring	✗	✗	✗	✗	(✓) (Bench-marking)	✗
Migration	✗	✗	✗ (virtual)	Offline, Ad-hoc	Ad-hoc	Online, Offline
Adaptable Topology	✗	✗	✗	✗ (Hadoop)	✗	✓
Automatic Replication	✗	✗	✗	(✓)	✗	✓
Automatic Partitioning	✗	✗	✗	✓	✗	✓
Logging + Recovery	✗	✗	✗	✗	✗	✗

	SQL++/ Forward	Cloud MdsQL	Estocada	Polybase	BigDAWG	Poly pheny DB
Data Models	NoSQL*, NewSQL, Relational	Relational, JSON-based, Array	JSON, Key-Value, Relational, XML	Relational	Array, Relational, Text	?
Common Data Model	JSON-based	JSON-based	Relational	Relational	✗	Assoc. Arrays
Annotations	✗	✗	✗	✗	✗	✓

* except graph & key-value

Summary

- Wide variety of use cases with diverse persistency requirements
- No „one fits all" solution
- Multiple ways of implementing intermediate layer
- A handful of solutions available (5+1 shown)
- Solutions are very rudimentary and tailored for very specific use cases
- We see a need for intelligent and feature-rich Multi-/Polystores

Conclusions for future Multi-/Polystores

- Able to adapt its data store topology
  - Provide "optimal" ecosystem for current requirements
- Routing data to "optimal" data store (Mediation)
  - based on user requirement, user behavior and queries
- Live Migration to adapt to changing topology + requirements
- Using flexible nature of the cloud to orchestrate adaptive topology

Our research focus

Thank you for your attention.

Sources

- *Tan, R.; Chirkova, R.; Gadepally, V.; Mattson,T. G. V.:* Enabling query processing across heterogeneous data models: A survey. In 2017 IEEE International Conference on Big Data (Big Data), 3211–3220 (2017)
- *Ong, K. W.; Papakonstantinou, Y.; Vernoux, R.:* The SQL++ query language: Configurable, unifying and semi-structured (2014)
- *Kolev, B.; Bondiombouy, C.; Valduriez, P.; Jiménez-Peris, R.; Pau, R.; Pereira, J.:* The CloudMdsQL Multistore System. In ACM SIGMOD, San Francisco, United States (2016)
- *Kolev, B.; Valduriez, P.; Bondiombouy, C.; Jiménez-Peris, R.; Pau, R.; Pereira, J.:* CloudMdsQL: Querying Heterogeneous Cloud Data Stores with a Common Language. Distributed and Parallel Databases, 34(4): 463–503 (2016)
- *Alotaibi, R.; Bursztyn, D.; Deutsch, A.; Manolescu, I.; Zampetakis, S.:* Towards Scalable Hybrid Stores: Constraint-Based Rewriting to the Rescue. In SIGMOD 2019 - ACM SIGMOD International Conference on Management of Data, Amsterdam, Netherlands (2019)
- *DeWitt, D. J.; Halverson, A.; Nehme, R.; Shankar, S.; Aguilar-Saborit, J.; Avanes, A.; Flasza, M.; Gramling J.:* Split query processing in polybase. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, SIGMOD ’13, pages 1255–1266, New York, NY, USA, ACM (2013)
- *Gadepally, V.; O’Brien, K.; Dziedzic, A.; Elmore, A.; Kepner, J.; Madden, S.; Mattson, T.; Rogers, J.; She, Z.; Stonebraker, M.:* Bigdawg version 0.1. In 2017 IEEE High Performance Extreme Computing Conference (HPEC), 1–7 (2017)
- *Yu, X.; Gadepally, V.; Zdonik, S.; Kraska, T.; Stonebraker, M.:* Fastdawg: Improving data migration in the bigdawg poly-store system. In Heterogeneous Data Management, Polystores, and Analytics for Healthcare, 3–15, Cham (2019)
- *Vogt, M.; Stiemer, A.; Schuldt, H.:* Polypheny-DB: Towards a distributed and self-adaptive polystore. In 2018 IEEE International Conference on Big Data (Big Data), 3364–3373 (2018)