Virtualizing Cloud Data Infrastructures with BRAD

Geoffrey X. Yu, Ziniu Wu, Ferdi Kossmann, Tianyu Li, Markos Markakis, Amadou Ngom
Tim Kraska, Samuel Madden

Design data infrastructures by focusing on application data use cases instead of physical infrastructure details

Data infrastructures are hard to design and change

- “One size does not fit all” led to a plethora of specialized cloud engines for data workloads
- Application code ends up tightly coupled to the physical infrastructure, making it hard to change

Virtual database engines (VDBEs)

Key idea: Capture the application-specific properties desired in a database engine

- 📝 Set of tables (with flag indicating writes)
- 📖 Query interface (SQL dialect, functionality required)
- 📈 Performance SLOs (e.g., p90 latency < 30 seconds)
- 📈 Freshness constraint (VDBE’s maximum staleness with respect to writes made to tables by any VDBE)

Key twist: The same table can be present in multiple VDBEs

Realizing VDBEs in BRAD

- BRAD implements VDBEs, working as middleware
- BRAD exposes an endpoint per VDBE, forwarding queries to physical engines
- 🔄 BRAD automatically realizes VDBEs onto physical engines using cost-based optimization with learned models
- 📊 Clients are decoupled from the physical infrastructure

BRAD optimizing data infrastructures

- BRAD scales down an overprovisioned infrastructure (Aurora and Redshift), maintaining SLOs while reducing cost
- Consolidates workload onto Aurora, shutting down Redshift
- Saves 4.6x in cost over next best baseline (TiDB), 6.0x in cost over starting physical infrastructure

---

dsg.csail.mit.edu/projects/brad

brad-project@csail.mit.edu