AnyDB: An Architecture-less DBMS for Any Workload

Tiemo Bang[1,2], Norman May[2], Ilia Petrov[3] and **Carsten Binnig**[1]

[1]Technical University Darmstadt [2] SAP SE [3]Reutlingen University

DATA MANAGEMENT
TU DARMSTADT

SAP

Hochschule Reutlingen
Reutlingen University

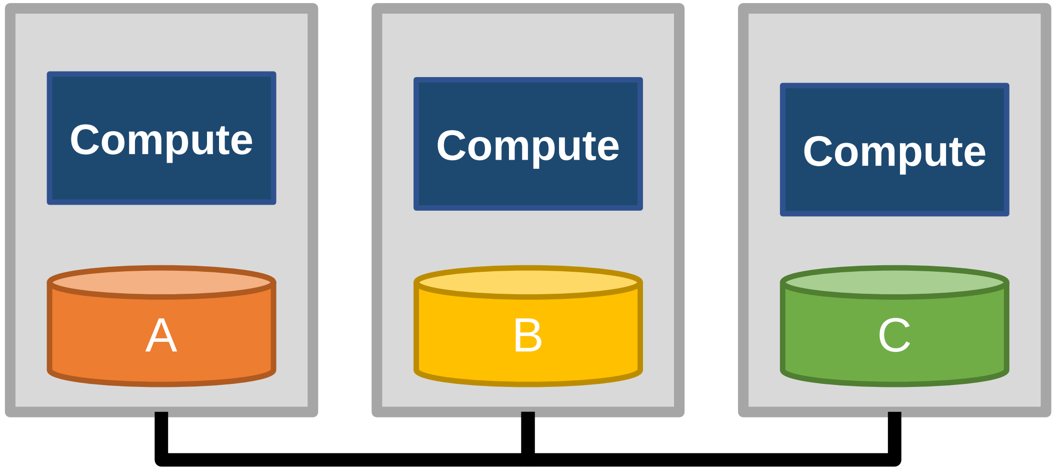
Scale-out DBMSs widely used today

Amazon Aurora
teradata.
Greenplum
ORACLE® TimesTen
SAP® HANA
amazon REDSHIFT
SQL
Azure SQL Database

Scale-out DBMSs are popular on-prem and in the cloud

Predominant scale-out DBMS Architectures

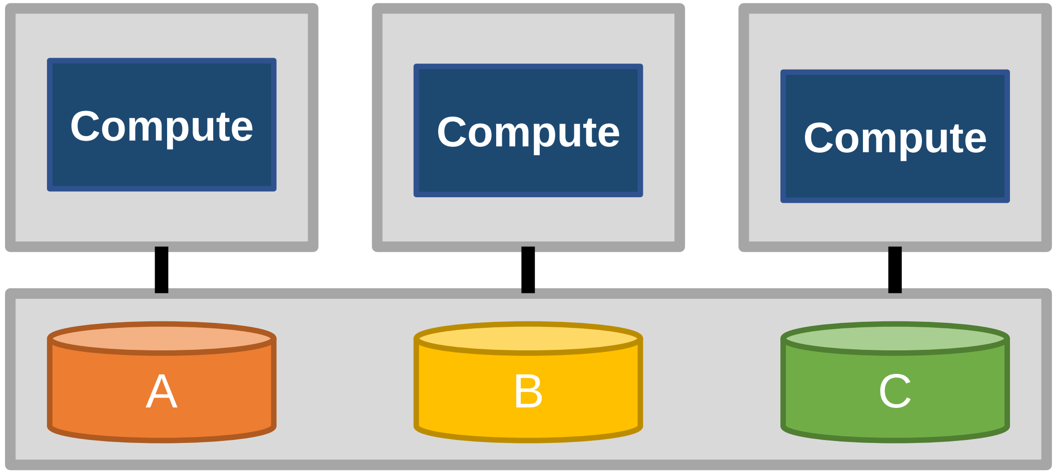
Shared-Nothing / Aggregated



+ Ideally suited for **partitionable** workloads

- Not optimal if load is non-uniform or quickly changes

Shared-Disk / Disaggregated



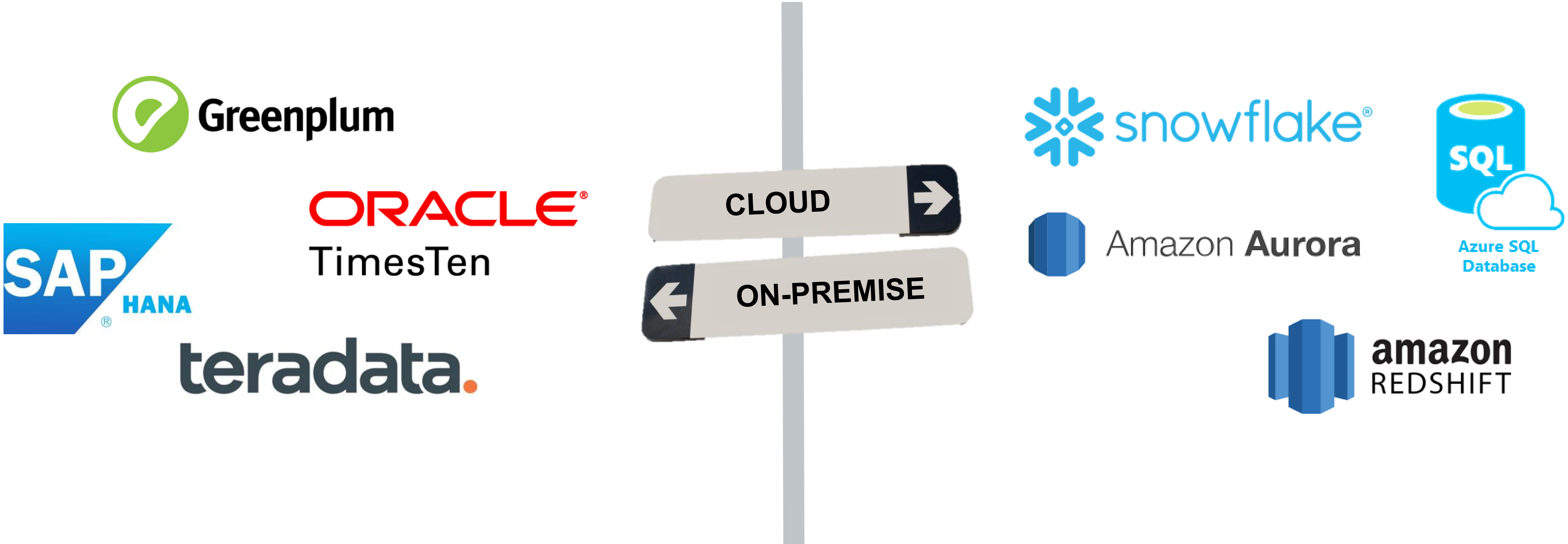
+ **Compute can be scaled elastically**

+ **More skew-tolerant**

- Higher latencies of data access → **caching needed**

Fate of DBMS determined by its architecture

Architecture of a DBMS is a design-time decision
→ many characteristics statically baked into DBMSs



Either a Shared-Nothing …

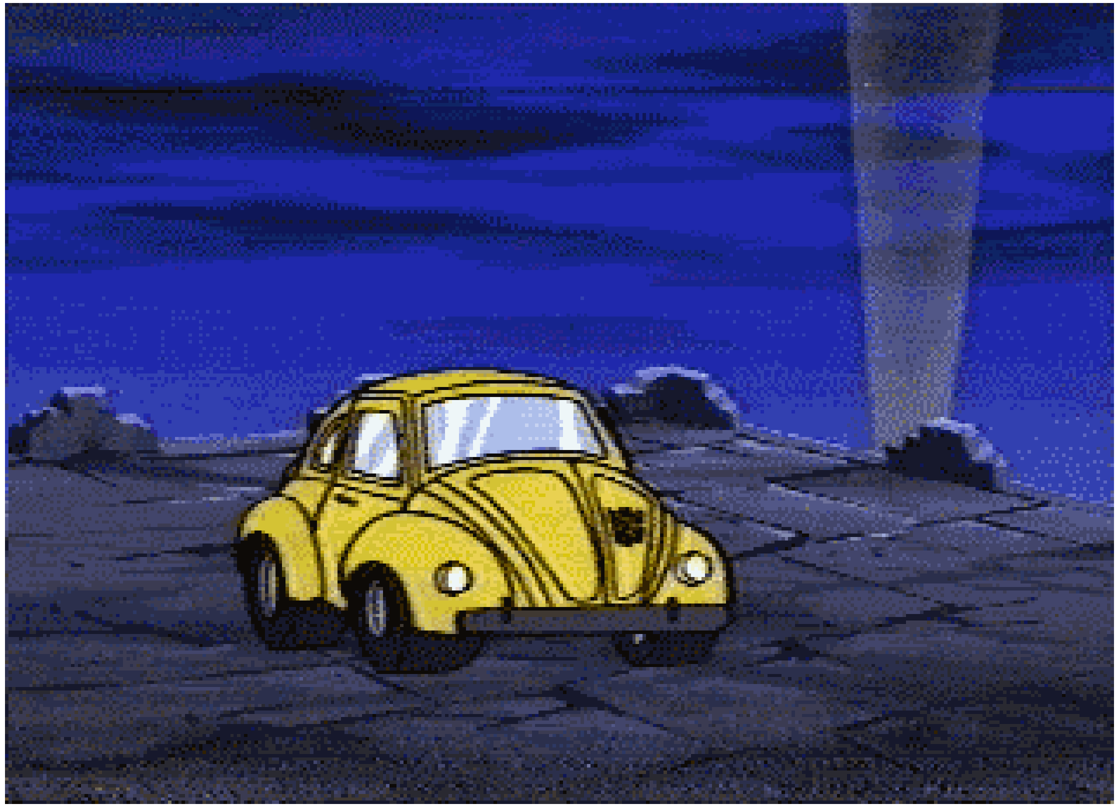
… OR a Shared-Disk DBMS

Designed for race tracks …

… but not for off-road!

Vision of AnyDB: An Architecture-less DBMS

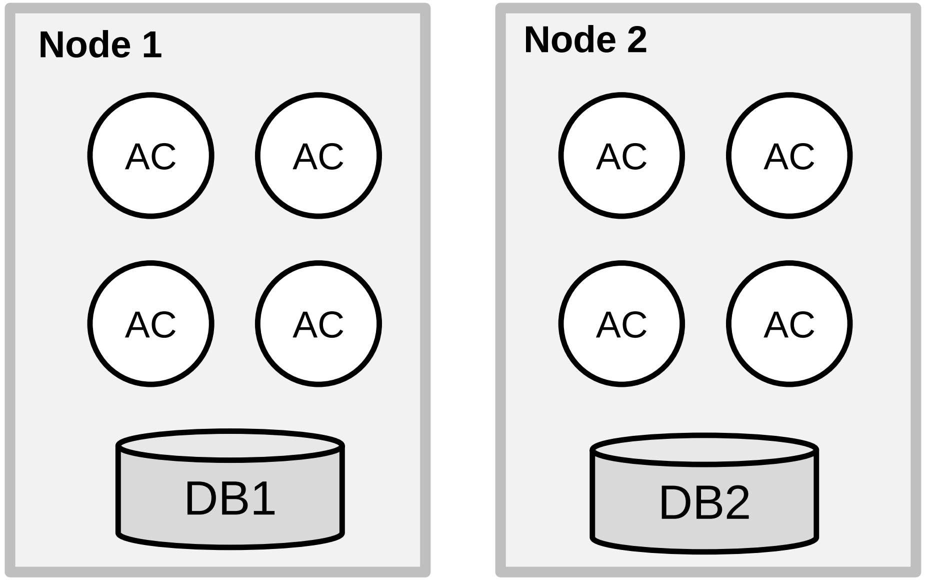
Defer choice of architecture to runtime!



→ **Mimic classical architectures** or **form new ones**
(e.g., a hybrid of Shared-Nothing and Shared-Disk)

Key Idea 1: No pre-defined Componentization

DBMS is composed of generic "AnyComponents" (ACs)
→ ACs can act as any DBMS component at runtime

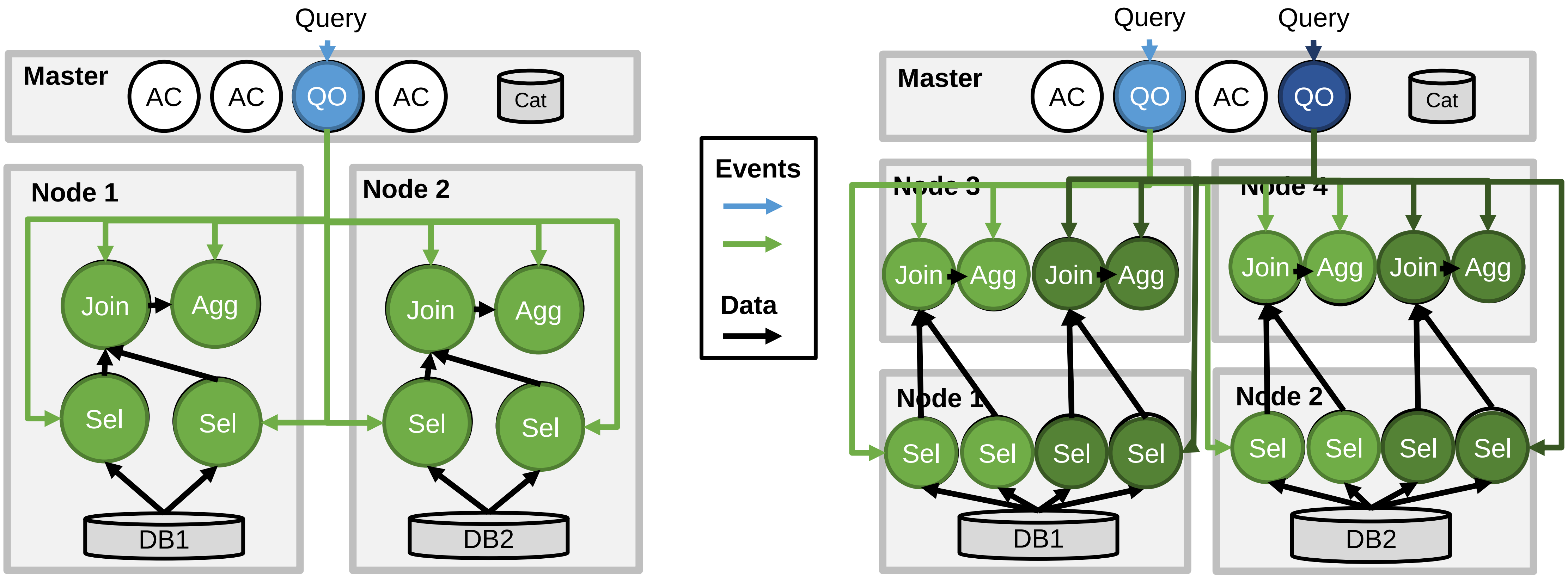


ACs are instrumented
by an **event & data stream**

Key Idea 2: Routing of Events & Data

Routing determines architecture at runtime (e.g., for OLAP)

OLAP (normal load): OLAP (increased load):



→ Shared-Nothing DBMS

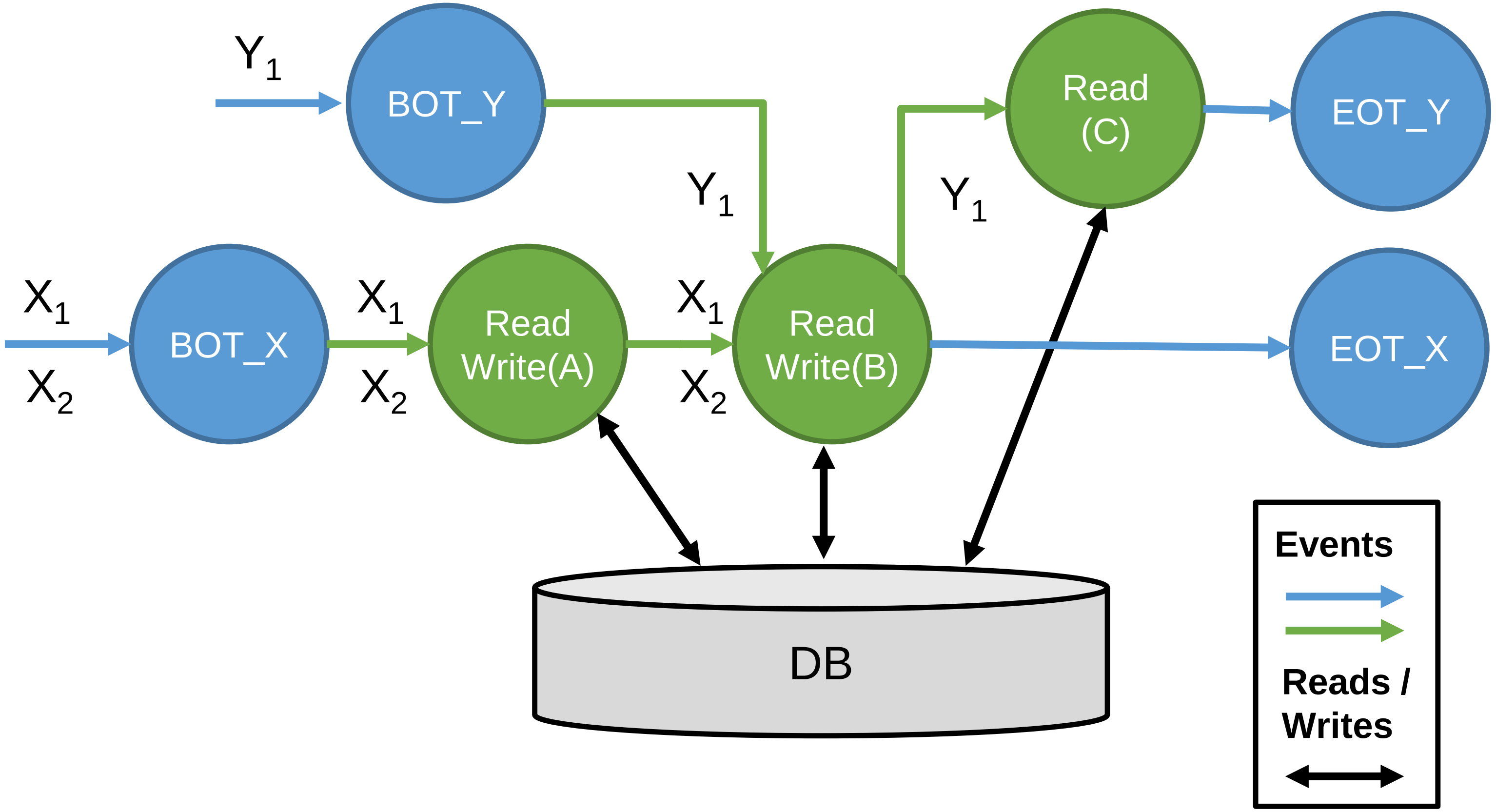
→ Shared-Disk DBMS (with pushdown)

Challenge: OLTP / Concurrency Control (CC)

Tx's can also be expressed as streams of events and data

```
Transaction X{
  Read(A);
  Write(A=A-50);
  Read(B);
  Write(B=B+50);
}

Transaction Y{
  Read(B);
  Read(C);
}
```



Challenge: OLTP / Concurrency Control (CC)

Tx's can also be expressed as streams of events and data

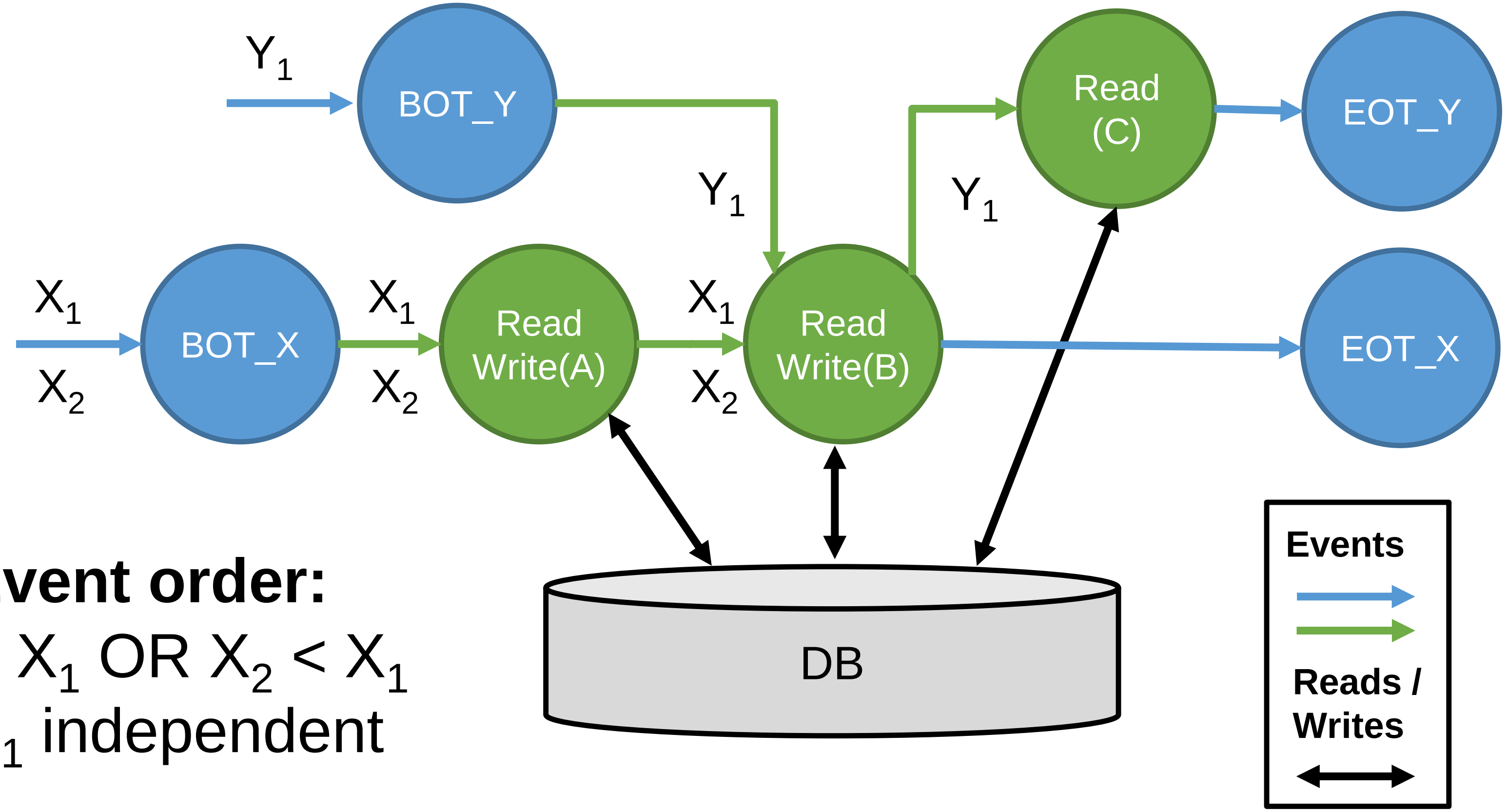
```
Transaction X{
  Read(A);
  Write(A=A-50);
  Read(B);
  Write(B=B+50);
}
```

```
Transaction Y{
  Read(B);
  Read(C);
}
```

Event order:

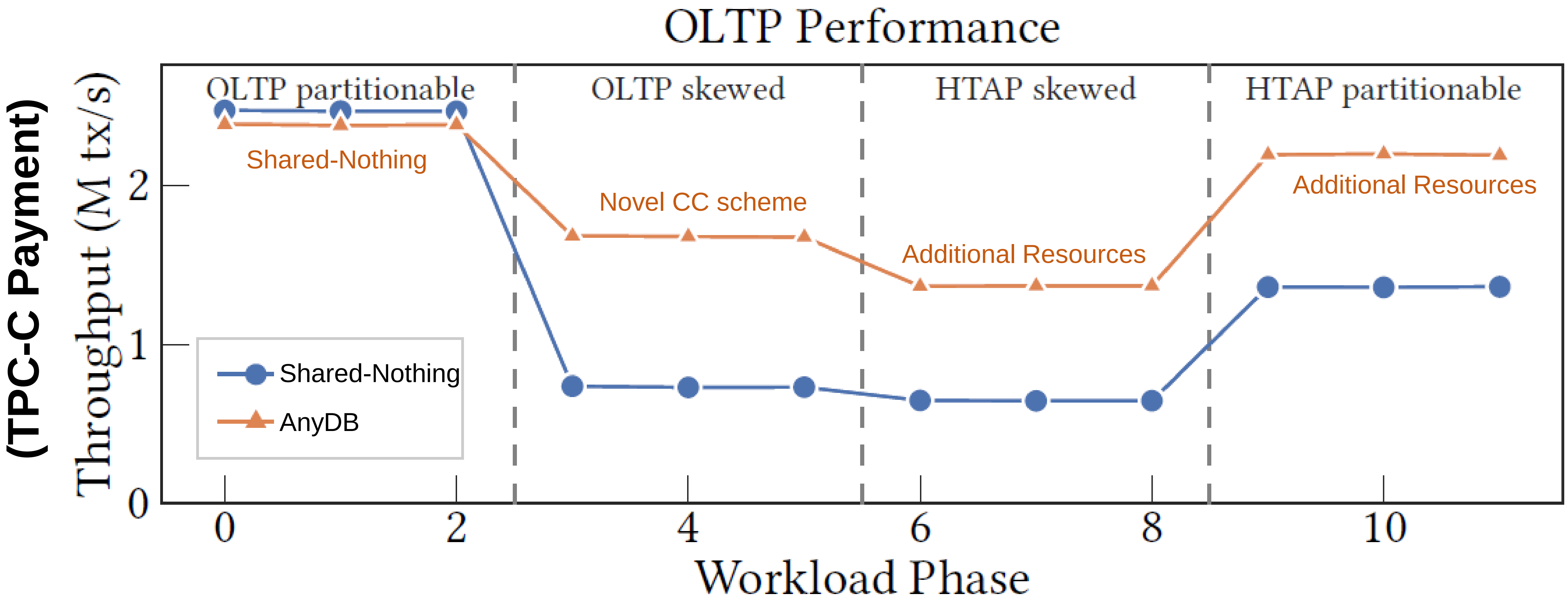
$X_2 > X_1$ OR $X_2 < X_1$

+ Y_1 independent



CC by event ordering → No distributed locking needed

Experiments: Initial Results



Results **confirm opportunities** of AnyDB for diverse workloads!

Summary & Future Directions

Vision of an Architecture-less DBMS: defer architectural decision to runtime

Many more details in the paper (e.g., efficient movement of state by pro-active data shipping called data beaming)

Future opportunities:

- Flexible routing opens up many other forms of adaption (e.g., to **include heterogenous compute resources on-the-fly**)
- Stateless execution of ACs is an **interesting alternative to build serverless-DBMSs** on top of FaaS

Thank you for your Attention!

AnyDB: An Architecture-less DBMS for Any Workload

Tiemo Bang[1,2], Norman May[2], Ilia Petrov[3] and **Carsten Binnig**[1]

[1)]Technical University Darmstadt [2)] SAP SE [3)]Reutlingen University

Challenge: Optimal Routing Decisions

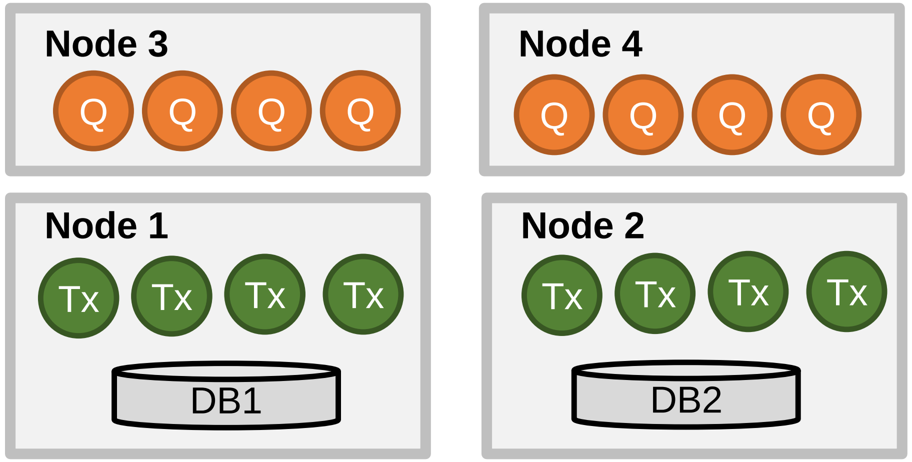
Optimizer needs to determine optimal routing based on worklaod

Two directions:

1. **Manual annotate with hints** (e.g. `EXECUTE … AS SHARED_NOTHING`)
2. **Automated routing** (e.g., learned approaches)

Example: Optimal Architecture → Hybrid

Sporadic OLAP → as Shared-Disk



Partitionable OLTP → as Shared-Nothing