Relational Cloud: A Database-as-a-Service for the Cloud

C. Curino, E. P. C. Jones, R. A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, N. Zeldovich
Database as a Service

• **Transactional, Relational DB Service**
  – hide complexity
  – exploit resource pooling
  – increase automation
  – (both for *private* and *public* cloud)
Existing Services

• Existing Commercial DB Services
  – Amazon RDS, SQL Azure (and many others)

• What they got right
  – simplified provisioning/deployment
  – reduced administration/tuning headaches

• What is still missing?
  – workload placement (to reduce hw cost)
  – automatic partitioning
  – Encryption (to achieve data privacy)
Relational Cloud Architecture

Figure 1: Relational Cloud Architecture.
Workload Placement

• Why
  – Load balancing
  – High Performance

• Problem Definition
  – Allocate workloads to servers in a way that
    1. minimizes number of servers used
    2. balances load across servers
    3. maintains performance
DBMS's tend to use all available resources.
Workload Placement

measure resource utilization

W1

W2

W3

disk i/o
ram
cpu

disk i/o
ram
cpu

disk i/o
ram
cpu

DBMS's tend to use all available resources

estimate combined load
numerical models

resource non-linearities

DBMS's tend to use all available resources. 

Note different slope for different working set sizes.
Workload Placement

- **measure resource utilization**
  - W1
  - W2
  - W3
  - Disk i/o
  - Ram
  - CPU

- **estimate combined load**
  - Numerical models
  - Note different slope for different working set sizes

- **find optimal assignment**
  - Non-linear programming
  - DBMS's tend to use all available resources
  - Resource non-linearities
  - Non-linear constraints and objective function

DBMS's tend to use all available resources

resource non-linearities

non-linear constraints and objective function
Partitioning

• Why
  – Scalability
  – Manageability

• Problem Definition
  – Partition the database into $N$ chunks in a way that maximizes the workload performance
Graph-based Partitioning

Input
Database

Workload Trace

Graph Representation

Explanation

Input (logs + preprocessing)

P: 0  ID < 4
P: 1  ID >= 4
Graph-based Partitioning

Input
- Database
- Workload Trace

Graph Representation
- Transaction edges
- Nodes: 1, 2, 3, 4, 5
- Edges: 1-2, 2-3, 3-4, 4-5
- Partitioning:
  - P: 0, ID < 4
  - P: 1, ID >= 4

Explanation

Input (logs + preprocessing)

Graph Partitioning (METIS)
Graph-based Partitioning

- Input (logs + preprocessing)
- Graph Representation
- Classification (Decision Tree)
- Explanation

- Input Database
- Workload Trace

- Graph Partitioning (METIS)

- Explanation:
  - P: 0, ID < 4
  - P: 1, ID >= 4
Encryption

• Why
  – Data Privacy

• Problem Definition
  – Minimize confidential info released to server ⇔ Efficiently execute queries
  – Minimize the amt of data leaked when application server is compromised
Onion Encryption

SELECT * FROM emp WHERE rank = 'CEO';
Onion Encryption

SELECT * FROM emp WHERE rank = ‘CEO’;

UPDATE table1 SET col1-OnionEq = Decrypt_RND(key, col1-OnionEq);

SELECT * FROM table1 WHERE col1-OnionEq = xda5c0407;
Conclusions

- Database as a Service has real potential
- Key Features to fully enable DBaaS
  - Workload Placement
  - Automatic Partitioning
  - Provable Privacy